2021 The 21st International Conference on Control, Automation and Systems (ICCAS 2021)
Ramada Plaza Hotel, Jeju, Korea, Oct. 12∼15, 2021

Vision-Based 3D Reconstruction Using a Compound Eye Camera

Wooseok Oh[1], Hwiyeon Yoo[1], Timothy Ha[1], and Songhwai Oh[1*]

[1]Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul, 08826, Korea,
wooseok.oh@rllab.snu.ac.kr, hwiyeon.yoo@rllab.snu.ac.kr, timothy.ha@rllab.snu.ac.kr, songhwai@snu.ac.kr

* Corresponding author

Abstract: The vision-based 3D reconstruction methods have various advantages and can be used in various applications such as navigation. Although various vision-based methods are being studied, it is difficult to reconstruct many parts at once with a general camera because of a small FOV. To solve this problem, we propose a coarse but lightweight reconstruction method using a camera with a unique structure called a compound eye with various advantages such as large FOV. In the process, we devise a network that performs depth estimation on a compound eye structure to obtain a depth image containing 3D information from an RGB image. We tested our methods by collecting data using a compound eye camera implemented in a Gazebo simulation and simulation scenes we created. As a result, our 3D reconstruction method using the data we collected and the confidence score from our depth estimation result, can capture the environment with a high probability of 97.51%.

Keywords: 3D Reconstruction, Compound Eye, Depth Estimation, Deep Learning

1. INTRODUCTION

Environment reconstruction is an important problem for mobile robots to accomplish tasks like navigation. Depending on the type of sensor used, most environmental restoration methods are classified into two types: LIDAR-based methods [1, 2] and vision-based methods [3–5]. The LIDAR-based methods have the advantage of high accuracy but have the disadvantages of dependence on the expensive sensor and limited in the LIDAR detection range. On the other hand, the vision-based methods using RGB cameras have a low price and no distance limitation, while accuracy is relatively low and sensitive to ambient light.

While 2D environment information is enough for simple navigation scenarios, some types of mobile robots, such as UAVs, require 3D environment information for navigation [6]. However, vision-based 3D reconstruction methods using ordinary camera have a weakness that much information cannot be obtained instantly due to the limited field of view (FOV). In this paper, we tackle the vision-based 3D reconstruction by using a unique camera structure called the compound eye camera, which has a large FOV.

The compound eye camera, which mimics the eyes of insects, is a structure in which many small-resolution single eye cameras are located on the surface of a hemisphere. Due to this unique structure, compound eye camera has various advantages such as a large FOV, low aberrations, and a large depth of field [7–9]. There have been several studies, such as ego-motion estimation [9] and semantic segmentation [8], using this structure of the compound eye. The tasks studied in previous studies require only 2D information, but in the case of 3D reconstruction, 3D information such as depth information is required. However, the depth sensor is more expensive than the RGB sensor, so it is inefficient to use the depth sensor for every single eye.

In order to address these problems, we propose a novel depth estimation network that suits for the unique structure of the compound eye images, which the traditional depth estimation methods cannot deal with. Furthermore, we propose a point cloud based 3D reconstruction method with a compound eye camera using single-eye-wise depth estimation results. The proposed method aims for a coarse 3D environment reconstruction that is lightweight, which is suitable for small robots with limited resources. In experiments in ROS simulation, we implemented a compound eye camera and indoor scene environment. In the test environment, the proposed method achieved 97.51% accuracy of 3D reconstruction.

The contributions of the paper are summarized as follows :

- We propose a depth estimation network that works on compound eye images.
- We propose a coarse but lightweight 3D point cloud based reconstruction method with the compound eye.
- We implement compound eye in simulation, enabling data collection and real-time simulation.

2. RELATED WORK

2.1 Deep Learning Applications with Compound Eye

There have been some studies dealing with various computer vision tasks by applying a deep neural network to a compound eye image [7–9]. [7, 8] designed the compound eye hardware prototype and applied a deep neural network to the compound eye image of the same structure. A network for objectness estimation on a compound image was proposed in [7], and a scheme for semantic segmentation with low complexity was devised in [8]. [9] proposed convolution neural network (CNN) based ego-motion classification algorithm in the same image structure. However, none of them used images obtained through compound eye cameras. They instead used 2D

Fig. 1. Examples of scenes in Gazebo simulator

RGB images converted into compound eye images.

2.2 Monocular Depth Estimation

Most of the recent monocular depth estimation algorithms are CNN-based [10–13]. Many studies have been improving performance using various network structures and losses. For example, in [10], two networks were used: a network that predicts a coarse depth map through a global feature and a network that refines a coarse depth map. In training, they used a scale-invariant loss that is independent of the global scale of the depth value. [12] proposed a network for depth estimation by combining features extracted from different scales. In addition, they used a combination of complementary losses, each obtained from the difference in depth, difference in gradient, difference in surface normal between estimated depth map and ground truth depth maps.

2.3 Vision Based 3D Reconstruction

There have been several studies of vision-based 3D reconstruction. Most vision based 3d reconstruction methods use depth image to obtain 3d information [3, 4]. [3] performed dense 3D surface reconstruction in real-time by fusing the depth data obtained while moving the Kinect sensor. Unlike [3], which can only be used in largely static environments, [4] enables 3D surface reconstruction in non-rigid scenes. In [4], the scene of each frame is transformed into the first frame through the motion field, and the transformed data is combined in the same way as in [3]. However, this method is not suitable for the reconstruction of large environments.

3. METHOD

3.1 Construct Simulation Environment

We constructed an environment that can collect compound eye data to learn depth estimation and test our 3D reconstruction method. In the simulation, we implemented a compound eye camera and created scenes similar to the real one. Unlike [7–9], which transformed a 2D RGB image into a compound eye image for network training, we collected images using a compound eye camera module in the simulation.

3.1.1 Compound eye camera in simulation

We implemented a compound eye camera on the Gazebo simulator [14]. The compound eye camera consists of several image sensors implemented in the simulator. Each image sensor, which we call single eye, receives an RGB-D image as input and has low resolution. The compound eye camera has a structure in which these single eyes are placed in multiple layers on the hemisphere's surface. There is one single eye in the center of the sensor in the first layer, and in the nth layer other than the first, $8n - 8$ single eyes are placed in a circular shape at regular intervals. The number of layers l and the resolution of a single eye can be changed in the simulation. We used compound eyes with 441 single eyes in 11 layers, and the size of each single eye image is 10×10 pixels, which shows better performance than other configurations for tasks in [7–9].

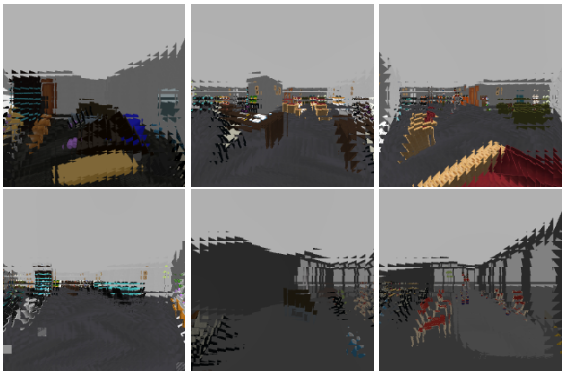


Fig. 2. Examples of Compound eye RGB images

3.1.2 Simulation scenes and compound eye data

We created simulation scenes on Gazebo to use our compound eye camera. Several objects such as desks and chairs were arranged to generate realistic scenes like office and cafe. We created nine scenes with sizes ranging from 10×10 m^2 to 20×20 m^2. Sample scenes are shown in Fig. 1.

Compound eye image $I_c \in \mathbb{R}^{21\times21\times3\times10\times10}$ and compound eye depth image $D_c \in \mathbb{R}^{21\times21\times1}$ were collected at each position by randomly changing the position of the compound eye camera on the environment. The examples of images are shown in Fig. 2. Each single eye image is partially overlapped with the adjacent single eye image.

3.2 Depth Estimation with Compound Eye Image

3.2.1 Network architecture

Similar to the previous studies about depth estimation [10–13], our network is CNN based. The difference is that each pixel of the output depth map represents the average depth value of the corresponding single eye view. Since we use one depth value for one single eye camera,

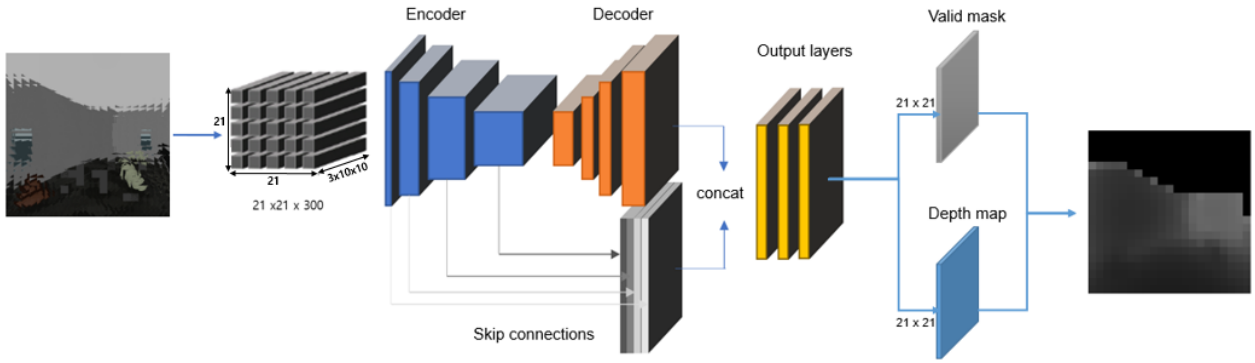


Fig. 3. Overview of our depth estimation network

our result is coarse compared to the depth estimation result using a typical image. Fig. 3 shows an overview of our network for depth estimation. The compound eye image data described in 3.1.2 is the network input. The network is composed of two parts; encoding and decoding parts. In the encoder, features of each single eye image are first extracted through 1×1 convolution layer, and then features at multiple scales are obtained using the ResNet18 [15] structure. In the decoder, the features from the encoder are upsampled to the depth map size using a convolution layer and bilinear upsampling. Also, through skip connections, features of multiple scales are upsampled to match depth map size and then concatenated with the features from the decoder. The final output of the network is computed from the integrated features.

The network's output consists of two components. One is a binary classification output that represents whether the depth value of each single eye is within the depth threshold d_{th}, and the other is a scalar output that estimates the depth value of each single eye view. Since the proposed method is designed for small robots and the estimation error increases as the distance increases, it is more effective to use only the depth values at a close distance. So, we improved the accuracy by using depth estimation only on single eye views classified as short distances. We used 4.5 m as d_{th}.

3.2.2 Loss function

For training our network, we used the loss L by adding the classification loss to the regression loss used in [12]. We define the Loss L between the estimated depth map d and the ground truth depth map g:

$$L(d, g) = L_{bce}(d, g) + \gamma L_{reg}(M(d), M(g)), \quad (1)$$

where L_{bce} is binary cross entropy loss function, γ is weight parameter and M defined as follows:

$$M(x) = \begin{cases} x, & \text{if } x < d_{th} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

We define L_{reg} as weighted sum of three loss functions:

$$L_{reg} = L_{depth} + \lambda L_{grad} + \mu L_{normal}, \quad (3)$$

where

$$L_{depth} = \frac{1}{n}\sum_{i=1}^{n} F(e_i), \quad (4)$$

$$L_{grad} = \frac{1}{n}\sum_{i=1}^{n}(F(\nabla_x(e_i)) + F(\nabla_y(e_i))), \quad (5)$$

$$L_{normal} = \frac{1}{n}\sum_{i=1}^{n}(1 - \frac{\langle n_i^d, n_i^g \rangle}{\sqrt{\langle n_i^d, n_i^d \rangle}\sqrt{\langle n_i^g, n_i^g \rangle}}), \quad (6)$$

where $F(x) = \ln(x + \alpha)$ and $e_i = \|d_i - g_i\|_1$. See Section 3.2 in [12] for more details regarding L_{reg}.

3.3 3D Reconstruction with Compound Eye Depth

Our algorithm performs 3D reconstruction using trajectory data consisting of a sequence of RGB-D compound images. We reconstruct the environment using only the part where the depth value was below the threshold and represented the reconstructed result in the form of a point cloud. The compound eye has a structure in which each single eye has a different direction and one depth value. We form a point cloud at the position represented by the direction and depth value of each single eye in one state of the trajectory. First, we consider the case when compound eye camera exists at the origin O along the x-axis. Let $(n+1)$th layer's ith single eye as $C_{n,i}$, the direction vector of $C_{n,i}$ as $u_{n,i}$, where $u_{0,0} = [1, 0, 0]^T$. In this case, $u_{n,i}$ is calculated as follws:

$$u_{n,i} = R_{xy}(\theta)R_{yz}(\alpha_n \cdot i) \cdot u_{0,0} \quad (7)$$

where

$$R_{xy}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_{yz}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix},$$

θ is the azimuth angle between $\overline{OC_{n,i}}, \overline{OC_{n+1,i}}$, and α_n is the difference of elevation angles between $\overline{OC'_{n,i}}, \overline{OC'_{n,i+1}}$ where $C'_{n,i}$ is the projected point of $C_{n,i}$ on the yz plane. Then, in the general case, we can calculate the point $x_{n,i}$ corresponding to $C_{n,i}$ as follows:

$$x_{n,i} = P_c + d_{n,i} \cdot R \cdot u_{n,i}, \quad \text{when } d_{n,i} < d_{th}. \quad (8)$$

where P_c is the position of the compound eye camera's center, $d_{n,i}$ is the depth value of $C_{n,i}$ and R is the rotation matrix of the compound eye camera.

To integrate our algorithm with the depth estimation method, we used a confidence score, representing the reliability of the reconstruction results. Since our depth estimation result is not perfectly accurate, there are errors in the reconstruction results using them. Therefore, the confidence score is updated while reconstruction along the trajectory, and points with a confidence score greater than one are considered valid observations. We use the probability p that the error of the depth estimation value is below a certain threshold to update the confidence score. In the confidence score update step, if the distance between the points of the newly added point cloud and the nearest previous point is less than 0.8 m, p is added to the confidence score of that point. We use KD-tree [16] to find the nearest point. For a detailed description of the algorithm, see Algorithm 1.

Algorithm 1 can reconstruct many areas at once by using the large FOV of the compound eye camera. Also, since we perform depth estimation in single eye wise, the size of the estimated depth map is smaller than that of a typical depth image.

Algorithm 1 3D reconstruction with depth estimation

Input: depth estimation network **N**, trajectory T = $\{(v_i, I_i)\}_{i=1}^{M}$ where v_i is position and rotation of camera, I_i is rgb compound eye image

Output: P_{output} : set of points

$pc = \emptyset$
for $i = 1 \rightarrow N$ **do**
  $P = depthtopoints(\mathbf{N}(I_i), v_i)$ ▷ using Eq. 7 and Eq. 8.
  $pc' = pc$
  for each $point \in P$ **do**
    if $i \neq 1$ **then**
      $idx, distance = KDtree(point, pc)$ ▷ find nearest neighbor
      if $distance < 0.8$ **then**
        $pc'[idx][1] \leftarrow pc'[idx][1] + p$ ▷ update confidence score
    $pc' \leftarrow pc' \cup \{(point, p)\}$
  $pc \leftarrow pc'$
$P_{output} = \emptyset$
for each $(point, score) \in pc$ **do**
  if $score \geq 1$ **then**
    $P_{output} \leftarrow P_{output} \cup \{point\}$

4. EXPERIMENTS

4.1 Dataset

Depth estimation and 3D reconstruction experiments were conducted in simulation. There are nine scenes we created in our simulation. Each was made to imitate cafe, kitchen, and office, as shown in Fig. 1. They has a size of about $10 \times 10\ m^2$ to $20 \times 20\ m^2$ We created scenes by placing objects so that the distance between them does not exceed 2 m.

For each scene, we collected data using the compound eye camera we implemented. A compound eye camera was randomly spawned, and 1000 images were collected from each scene to collect 9000 images. Among them, in 4 scenes, 12 trajectories consisting of 100 images were collected in consecutive positions while moving the camera slightly. We trained a depth estimation network using 7000 images and tested it on other 2000 images. The trajectory data were used to test the 3d reconstruction.

4.2 Performance Comparison

4.2.1 Depth estimation

We compared our trained depth estimation network with the mean depth image computed across the training set. We evaluated each method using a measurement called threshold accuracy, which is as follows:

$$\%\ \text{of}\ d_i\ \text{s.t}\ \max(\frac{d_i}{g_i}, \frac{g_i}{d_i}) = \delta < \delta_{thr}. \tag{9}$$

As in many other papers, we compared the performance using three values of δ_{thr}: 1.25, 1.25^2, and 1.25^3.

4.2.2 3D reconstruction

To check the accuracy of our reconstruction method using the estimated depth map, we compared the results reconstructed using the ground truth depth with the results reconstructed using estimated depth. We used a percentage of $\delta < 1.25$, a depth estimation performance index, as a confidence score p to reconstruct from the estimated depth map. The two reconstruction results are represented as a point cloud, called pc_{gt} and pc_{est}, respectively. We used two evaluation criteria: the first is the proportion of the pc_{est} that have a matched point in pc_{gt}, and the second, conversely, is the proportion of the pc_{gt} that have a matched point in pc_{est}. They are called Precision and Recall, respectively. These two measurements provide an overview of how well our method reconstructed the environment. In the experiments, we defined that the two points are matched if the distance between them is less than 0.8 m.

5. RESULT

5.1 Depth estimation

We compared our method with the mean depth image computed in the training set, which is the lower bound performance. The depth estimation results for our collected data set are shown in Table 1. Our method showed 69.8% performance for $\delta < 1.25$, which is about 10% higher than the mean depth image. In addition, it showed high performance as 91.4% for $\delta < 1.25^3$, and high classification accuracy as 92.4%.

The depth image estimated by our method and the ground truth depth image can be seen in Fig. 4. The estimation results of our method mainly estimate the depth values of walls and floors well. However, the accuracy

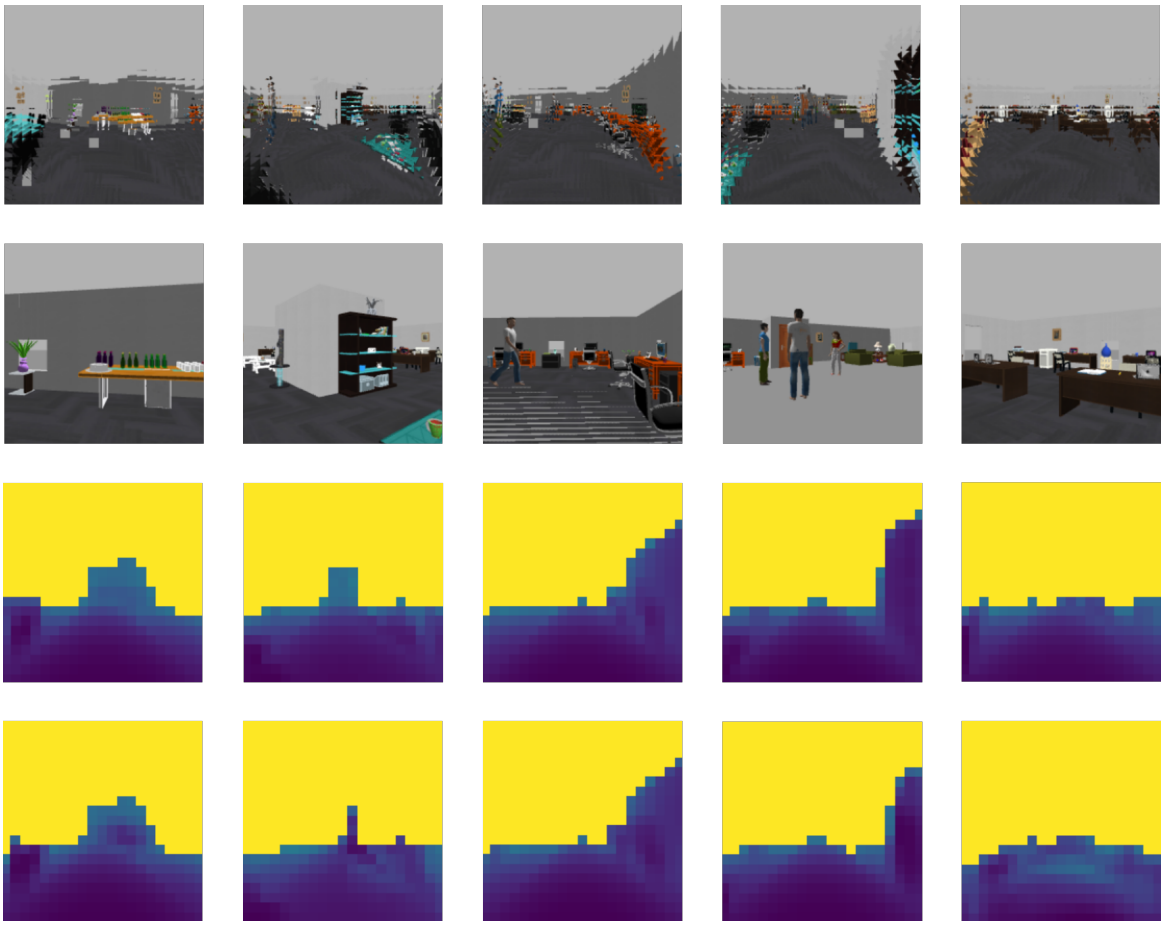


Fig. 4. Results of depth estimation with compound eye images. From first to last row; Compound eye RGB images, typical RGB images at the same position, ground truth compound eye depth images, estimated depth images using our method. In the depth images, the yellow area is the area where the depth value exceeds d_{th}.

Table 1. Comparison of depth estimation results on our compound eye image dataset

	δ <1.25	δ <1.25^2	δ <1.25^3	Classification Accuracy
Ours	69.8%	84.2%	91.4%	92.4%
Mean	58.7%	80.2%	86.6%	-

Table 2. Recall and Precision of our 3D reconstruction method

	Recall	Precision
Ours $\leftrightarrow$ gt	97.51%	98.09%

of other objects such as tables is relatively low. It can be seen that the classification accuracy is high, and most misclassification is made at the boundary where the depth value is near d_{th}.

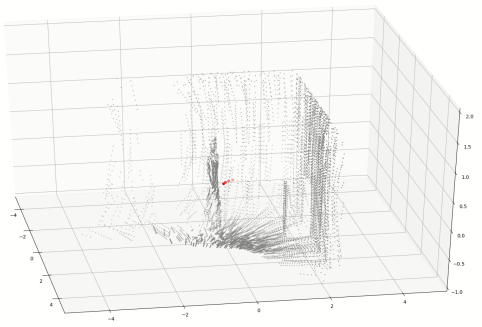
5.2 3D reconstruction

We compared the results of our reconstruction method using depth estimation pc_{est} with reconstruction using ground truth depth pc_{gt}. We use the probability of $\delta < 1.25$ of the depth estimation network (0.698) as p in Algorithm 1. As provided in Table 2, our method shows high Recall and high Precision, which means that pc_{est} contains most of pc_{gt}, and also most of pc_{est} correctly reconstructed the scene. From this result, the task such as collision avoidance can perform well using the reconstruction results of our method.

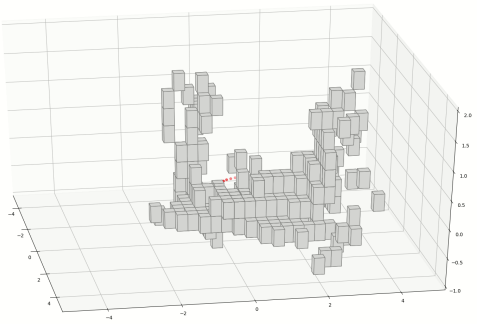
The results of 3D reconstruction pc_{gt} and pc_{est} are shown in Fig. 5. In Fig. 5, (a) is the result of reconstruction using ground truth depth without d_{th}, and (b) is the result of reconstruction using estimated depth map with d_{th}. Because we use the probability of $\delta < 1.25$, we visualize pc_{est} using voxels instead of points. We use voxels with edges of length 0.3 m in Fig. 5 (b). It can be seen that the reconstruction result using the estimated depth map is similar to pc_{gt} even though a small voxel is used.

6. CONCLUSION

We proposed a method for depth estimation using this data and 3d reconstruction method with a compound eye using an estimated depth map. We also implemented a compound eye camera and collected data to evaluate the methods in the simulation. Our proposed method is able to reconstruct the simulation environment accurately, and



(a)



(b)

Fig. 5. (a) Result of reconstruction using ground truth depth. (b) Voxel representation of reconstruction result with estimated depth map.

it is expected that our method can be used when performing tasks such as collision avoidance. Above all, our method is simple, so it is thought to be suitable for hardware with low computing power. In addition, since we have implemented an environment that can experiment in the simulation, so if the compound eye camera is implemented as hardware, it is expected that research such as learning from simulation and transfer to the real environment will be possible.

7. ACKNOWLEDGMENT

This research was supported by a grant to Bio-Mimetic Robot Research Center Funded by Defense Acquisition Program Administration, and by Agency for Defense Development (UD190018ID).

REFERENCES

[1] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *Robotics: Science and Systems, RSS*, D. Fox, L. E. Kavraki, and H. Kurniawati, Eds., 2014.

[2] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3d lidar-based online mapping," in *IEEE International Conference on Robotics and Automation, ICRA*, 2018.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality, ISMAR*, 2011.

[4] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015.

[5] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[6] Y. Lu, Z. Xue, G. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo spatial Inf. Sci.*, vol. 21, no. 1, pp. 21–32, 2018.

[7] H. Yoo, D. Lee, G. Cha, and S. Oh, "Estimating objectness using a compound eye camera," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI*, 2017.

[8] G. Cha, H. Yoo, D. Lee, and S. Oh, "Light-weight semantic segmentation for compound images," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI*, 2017.

[9] H. Yoo, G. Cha, and S. Oh, "Deep ego-motion classifiers for compound eye cameras," *Sensors*, vol. 19, no. 23, p. 5275, 2019.

[10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Conference on Neural Information Processing Systems, NIPS*, 2014.

[11] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *CoRR*, vol. abs/1812.11941, 2018.

[12] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2019.

[13] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *International Conference on 3D Vision, 3DV*, 2016.

[14] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2004.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001.