IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2020, accepted February 24, 2020, date of publication February 28, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977109

SelfSphNet: Motion Estimation of a Spherical Camera via Self-Supervised Learning

DABAE KIM, (Student Member, IEEE), SARTHAK PATHAK, (Member, IEEE), ALESSANDRO MORO, ATSUSHI YAMASHITA, (Member, IEEE), AND HAJIME ASAMA, (Fellow, IEEE)

Department of Precision Engineering, The University of Tokyo, Tokyo 113-8656, Japan

Corresponding author: Dabae Kim (kimdabae@robot.t.u-tokyo.ac.jp)

This work was supported by the Japan Society for the Promotion of Science, Grant-in-Aid for JSPS Fellow (KAKENHI), under Grant 18F18109.

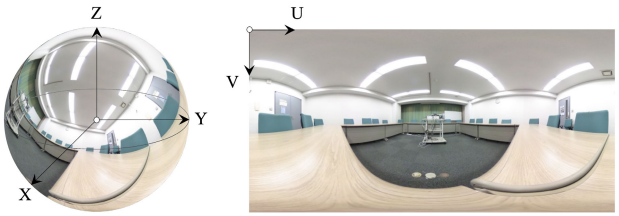
ABSTRACT In this paper, we propose SelfSphNet, that is, a self-supervised learning network to estimate the motion of an arbitrarily moving spherical camera without the need for any labeled training data. Recently, numerous learning-based methods for camera motion estimation have been proposed. However, most of these methods require an enormous amount of labeled training data, which is difficult to acquire experimentally. To solve this problem, our SelfSphNet employs two loss functions to estimate the frame-to-frame camera motion, thus giving two supervision signals to the network with the usage of unlabeled training data. First, a 5 DoF epipolar angular loss, which is composed of a dense optical flow of spherical images, estimates the 5 DoF motion between two image frames. This loss function utilizes a unique property of the spherical optical flow, which allows the rotational and translational components to be decoupled by using a derotation operation. This operation is derived from the fact that spherical images can be rotated to any orientation without any loss of information, hence making it possible to ''decouple'' the dense optical flow between pairs of spherical images to a pure translational state. Next, a photometric reprojection loss estimates the full 6 DoF motion using a depth map generated from the decoupled optical flow. This minimization strategy enables our network to be optimized without using any labeled training data. To confirm the effectiveness of our proposed approach (SelfSphNet), several experiments to estimate the camera trajectory, as well as the camera motion, were conducted in comparison to a previous self-supervised learning approach, SfMLearner, and a fully supervised learning approach whose baseline network is the same as SelfSphNet. Moreover, transfer learning in a new scene was also conducted to verify that our proposed method can optimize the network with newly collected unlabeled data.

INDEX TERMS Motion estimation, computer vision, image processing, deep learning, convolutional neural networks.

I. INTRODUCTION

Motion estimation of cameras in frame sequences is a critical part of robotic applications such as visual odometry [1], [2] and structure from motion [3], [4], as they require ego-motion of cameras. Especially, visual Simultaneously Localization and Mapping (SLAM) techniques [5] are dependent on the camera motion estimation. Various types of cameras have been adopted to track the motion of camera-equipped autonomous vehicles or drones by using consecutive images in all frames. Perspective cameras, which project images on a plane, are popular for numerous applications. However, these cameras often suffer from their narrow field-of-view, which may fail to track the camera motion due to occlusions like moving objects [6]. They can also lose their location due to images with fewer features such as textureless regions or shadows. Meanwhile, spherical cameras (Fig. 1), which have a wider 360° field-of-view than that of perspective cameras, are relatively beneficial for camera motion estimation [7], [8], as they are less vulnerable to environments with fewer features, occlusions, moving objects, and other problems. Multiple studies have explored the utilization

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.



(a) Spherical (b) Equirectangular

FIGURE 1. Spherical projection. Spherical cameras can capture images in (a) spherical projection and (b) equirectangular projection without any loss of information because of their a 360° field-of-view.

of the all-round view of spherical cameras. Reference [8] rotated spherical images to avoid the distorted areas, and extracted robust features from the central region of the image. Reference [9] rectified the optical flow in an equirectangular projection by rotating the images on a sphere to estimate the motion of the spherical camera. These methods are based on the fact that spherical images contain all-round information and can be rotated without any loss of information.

Modeling camera motion estimation is an arduous problem. Learning-based approaches to estimate the motion of spherical cameras have recently explored the trajectories of cameras [10], [11]. Convolutional Neural Networks (CNNs) have been used for robust estimation in various environments and the accuracy is equivalent to that of traditional feature-based approaches. However, most learning-based approaches still require a large amount of labeled training data, which is difficult to acquire and label. These are known as fully supervised learning approaches [12]. Assuming that the labeled data is captured, the training might lead to overfitting in some specific scenes. This indicates the accuracy in test scenes cannot be guaranteed. In contrast, self-supervised learning approaches for deep learning tasks have recently attracted much attention [13]–[17]. These approaches do not require any explicitly labeled data for training, and thus present the possibility of optimizing the network using unlabeled data. They have been adopted in various regression tasks, such as depth estimation [18], semantic segmentation [19], and motion estimation [13]. For these tasks, it is difficult to acquire precise ground-truth labels.

Self-supervised learning methods can easily retrain networks in a completely different environment with the use of newly collected training data whose labels are not provided. This retraining cannot be conducted by fully supervised learning methods because they require the labels of the training data. In this research, we conducted this retraining with a network whose weights were pre-trained by the transfer learning approach. This method only requires the collection of data, and does not need any additional pre-processing or labeling. By conducting this transfer learning, the estimation accuracy is expected to improve.

For optimizing self-supervised networks, extraction of the meaningful features of input data is required. To realize this, the network should be provided with pseudo-supervision signals instead of direct supervision signals, which are composed of camera motion labels. In this research, we construct the required supervision signals from the input data and loss functions. These signals are unique to the spherical camera. Furthermore, we train the spherical camera motion with the unlabeled data using our self-supervised learning approach.

Dense optical flow [20], [21], which represents pixel-wise movements, is often adopted to estimate the frame-to-frame camera motion. The advantage of this optical flow is that it is relatively less vulnerable to raw RGB intensities, which causes overfitting when training [22]. Furthermore, in spherical cameras, the optical flow can be distinguished easily between pure rotational and translational states (Fig. 3), as the optical flow on a sphere can be derotated using a simple multiplication of the rotation matrix. Consequently, the network can efficiently decouple the two components of the optical flow. In this research, we utilize this derotation operation to decouple the optical flow, which is required for composing a unique geometric loss function similar to that used in [23]. Therefore, we adopt this loss function for 5 DoF motion estimation using the first two consecutive frames, and then estimate the full 6 DoF motion using the next frames.

To estimate the full 6 DoF camera motion including a translation scale, we composed a photometric reprojection error using spherical warping, which generates synthetic spherical images. As spherical cameras have an all-round view without any loss of information, a lossless warping of synthetic images can be achieved by using depth maps and original images. However, a monocular camera does not have a standard for the translation scale, such as the baselines in stereo camera systems, when generating the depth map. To address this problem, we assign a constant value to fix the translation scale of $I_t \rightarrow I_{t+1}$, and then generate the depth map by using triangulation. This process estimates the full 6 DoF motion between the first and third frames $I_t \rightarrow I_{t+2}$ as a relative translation of the fixed scale.

To summarize, two loss minimizations are conducted to estimate the full 6 DoF motion of the spherical camera. First, the epipolar angular error minimization estimates the 5 DoF motion, which are the rotation and the translation direction. Next, the photometric reprojection error estimates the full 6 DoF motion, using the disparity map that is equal to the magnitude of the derotated (translation only) optical flow.

In this manner, we process three frames at a time to estimate the camera motion and trajectory. These two loss minimizations are carried out simultaneously by our SelfSphNet, without using labeled training data. An overview of our SelfSphNet is summarized in Fig. 2. Our network is divided into two parts, *Stream A* and *B*. In *Stream A*, a dense optical flow enters the CNN and the 5 DoF camera motion is obtained by minimizing the 5 DoF epipolar angular error. Here, the depth map can also be obtained by the decoupled optical flow. In *Stream B*, raw images enter the CNN and the full 6 DoF camera motion is obtained by minimizing the photometric reprojection error. The entire process can be done without using any labeled training data. To confirm the effectiveness

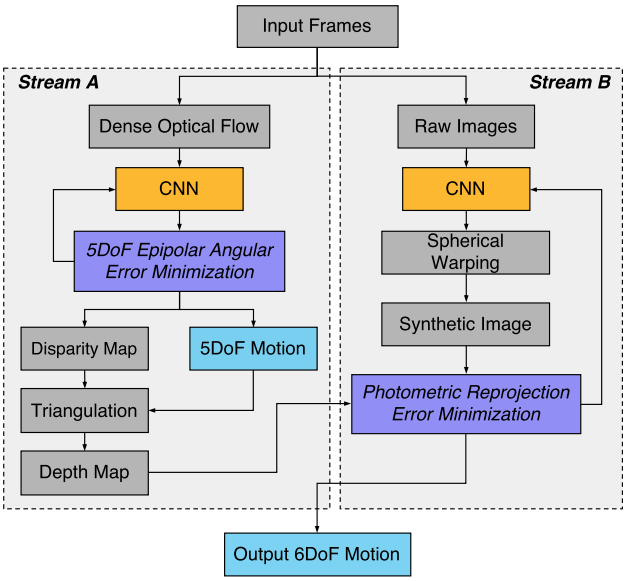


FIGURE 2. Overview of our proposed SelfSphNet. The input consists of the dense optical flow and raw images from a frame sequence and the output is the full 6 DoF motion of the spherical camera. Two loss minimizations (*Stream A* and *B*) are conducted in order to optimize both convolutional neural networks, and these are done by a self-supervised manner without any labeled training data.

of our SelfSphNet, the camera trajectory estimation is conducted using the estimated frame-to-frame camera motion by carrying out several experiments. For comparison with the fully supervised learning method, we additionally train our SelfSphNet, this time with the labeled training data, and confirm the possibility of the self-supervised camera motion estimation. The contributions of our research are as follows:

- We constructed a self-supervised learning network (SelfSphNet) to estimate the motion of a spherical camera without using any labeled training data.
- Spherical optical flow-based derotation enabled the network to estimate the depth map, thus avoiding overfitting, a problem methods based on raw images often encounter.
- Transfer learning, using the newly captured unlabeled data, was conducted to learn the structure of a previously unseen scene. The results we obtained with SelfSphNet for the camera motion estimation were more accurate compared to those obtained using fully supervised learning.

II. RELATED WORK

To expand the versatility of the camera-based motion estimation, omnidirectional [24], fisheye [25], or spherical cameras [26] have been adopted in recent years. They have proven to be more beneficial compared to perspective cameras, owing to their wider field-of-view. Considering real environments, camera-based motion estimation approaches often suffer from interruption caused caused by moving objects [27], partially occluded regions [28], and the lack of corresponding information between the frames [29]. Thus, it is essential to acquire a larger field-of-view to overcome the problems caused by these unexpected scenarios. Among the various large field-of-view cameras, the spherical camera, which consists of two fisheye lenses, has a full 360° field-of-view. Therefore, there exists plenty of potential for utilization of this unique device. Despite its wide field-of-view, there are disadvantages: the two fisheye lenses require complicated calibration and the projected images are distorted. However, we use spherical cameras that have already been calibrated and design a distortion weight to address the distortion problem.

Numerous self-(un)supervised visual odometry approaches have displayed the possibility of learning the network using unlabeled training data [13], [14], [18], [30]–[32]. These approaches attempted to provide supervision signals to train their networks using pseudo-labels generated from unlabeled training data. To obtain the required supervision signals, [33] generated synthetic images by using an image warping, [34] geometrically constrained a transformation matrix along multiple frames, and [35] introduced pose consistency, which is unique to a cubemap projection. However, to the best of our knowledge, only a few studies concerned with the self-supervised motion estimation of spherical cameras, such as [35], have been reported to date. Moreover, datasets suitable for a spherical camera are lacking. This prompted us to manually collect data to build datasets to enable the spherical camera motion to be learned and to construct the self-supervised learning network (SelfSphNet).

Recently, multiple studies have attempted to process spherical signals directly. For example, spherical convolution [36] and spectral analysis-based convolution [37], [38]. However, all these approaches are in their infancy and are capable of solving simple classification or recognition tasks rather than complex regression problems such as depth/motion estimation [31], [33] or camera relocalization [39]. In this research, we attempt to compose the unique loss functions using a spherical camera geometry, and minimizing them to estimate the full 6 DoF motion of the monocular spherical camera. These loss functions can optimize the network to solve the regression problem in a self-supervised manner.

To estimate the consecutive trajectory of a camera, it is essential to first acquire a depth map along the frame sequences. This is often calculated via triangulation that utilizes either the baselines of stereo camera systems or certain particular objects whose real sizes are already known. However, monocular camera systems have no such baseline, and the sized objects have a constraint in that they should be tracked in all frame sequences, which complicates the problem setting. Therefore, in many self-supervised approaches of the motion estimation for monocular cameras, a single raw image is often used to generate the depth map by using encoder-decoder networks such as those used in [18], [31], [33]. They perform well in trained environments, whereas this may not be realized in new environments because of overfitting problems. This is because these networks generate the depth map from a single image [18], [31], [33]. To solve

this problem, we use optical flow vectors that contain the motion information between two image frames rather than the single raw image. These optical flow vectors are less vulnerable to overfitting because they represent pixel movement, which is not affected by RGB values. This approach enables the network to be trained in more general conditions and to estimate the camera motion more robustly in various environments.

III. PROPOSED METHOD

Our proposed method to estimate the motion of a spherical camera utilizes two error functions, which are unique to spherical images. First, an epipolar angular error, which is used for estimating the 5 DoF camera motion of the first two consecutive frames $I_t \rightarrow I_{t+1}$, is explained using a derotation operation. Next, the disparity, which corresponds to the magnitude of the derotated optical flow, is converted into a depth map by using triangulation. In this triangulation, the translation between I_t and I_{t+1} is fixed to 1 without loss of generality, as the monocular camera lacks a baseline. Finally, a photometric reprojection error is introduced to estimate the full 6 DoF camera motion of $I_t \rightarrow I_{t+2}$, using the depth map generated from the triangulation.

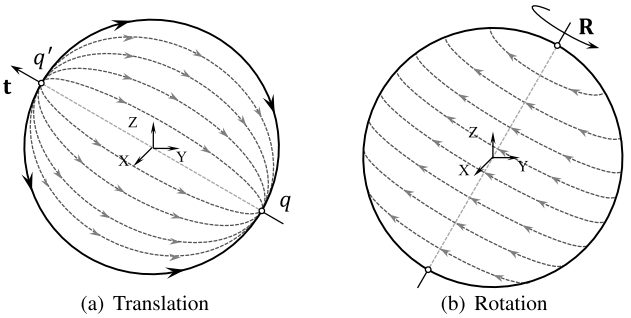
A. 5 DOF EPIPOLAR ANGULAR ERROR

Dense optical flow in consecutive frames represents the camera motion intuitively. When the camera moves, the optical flow shows different patterns as mixtures of translational and rotational components of the motion. In perspective cameras, the optical flow of the translation to the left side and the yawing rotation in the counterclockwise direction are similar. This confuses the network when estimating them separately. Meanwhile, the optical flow patterns obtained from spherical cameras are found to have a distinguishing property [40] for rotational and translational camera movements, as shown in Fig. 3. In a pure rotational state, the optical flow vectors move in circles perpendicular to the axis of the rotation (roll, pitch, and yaw), whereas, for a pure translational state, the optical flow vectors diverge from the epipoles q' and converge to the opposite epipoles q, and their alignment to the epipolar circles remain constant. This property enables the translational and rotational components of the optical flow $\mathbf{f}$ to be decoupled from the mixture, $\mathbf{f} = \mathbf{f}_{\text{rot}} + \mathbf{f}_{\text{trans}}$. This can be achieved using the derotation operation of the optical flow vectors, as shown in Fig. 6. After derotation, only the translational optical flow vectors remain, $\mathbf{f} = \mathbf{f}_{\text{trans}}$, which point towards the same directions as the epipolar circles.

In this case, the normal vectors of the derotated, *i.e.*, translational, optical flow $\mathbf{N}_{\text{f}}$ over all spherical unit vectors $\hat{\mathbf{x}} = [x, y, z]^T$, can be expressed as the cross product $\times$ of the location of the optical flow vectors $\hat{\mathbf{x}} + \mathbf{f}$ and $\hat{\mathbf{x}}$, on the unit sphere:

$$\mathbf{N}_{\text{f}} = (\hat{\mathbf{x}} + \mathbf{f}) \times \hat{\mathbf{x}}. \tag{1}$$

In a similar way, the normal vectors of the epipolar circles $\mathbf{N}_{\text{q}}$, regarding to the epipole $\mathbf{q}(\theta, \phi)$ can be defined as



(a) Translation (b) Rotation

FIGURE 3. Spherical optical flow. Optical flow on a unit sphere in the pure (a) translational and (b) rotational states. They can be decoupled using a derotation operation, which enables the network to estimate the camera rotation and translation separately.

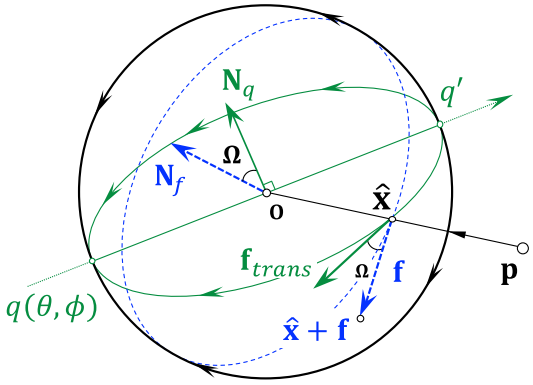


FIGURE 4. 5 DoF epipolar angular error. The cross products $(\hat{\mathbf{x}} + \mathbf{f}) \times \hat{\mathbf{x}}$ and $\mathbf{q} \times \hat{\mathbf{x}}$ are taken to find the angle Ω, between $\mathbf{f}_{\text{trans}}$ and $\mathbf{f}$ at $\hat{\mathbf{x}}$, projected from point p. According to the derotation of the rotational optical flow, two large circles are overlapped, minimizing the angular error Ω to be zero.

the cross product of $\mathbf{q}$ and $\hat{\mathbf{x}}$ on the unit sphere:

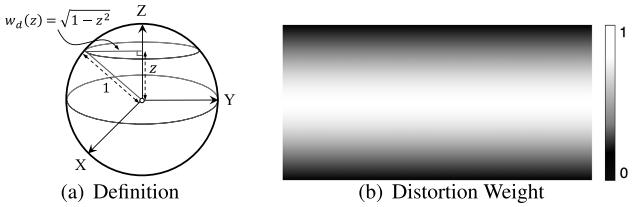
$$\mathbf{N}_{\text{q}} = \mathbf{q} \times \hat{\mathbf{x}}. \tag{2}$$

The directions of the above two normal vectors are required to be the same in a completely derotated state. Therefore, the 5 DoF epipolar angular error $\boldsymbol{\Omega}$ [23] is defined as the angular distance between $\mathbf{N}_{\text{f}}$ and $\mathbf{N}_{\text{q}}$ of $\hat{\mathbf{x}}$ over a unit sphere $\mathbb{S}$ (Fig. 4):

$$\boldsymbol{\Omega} = \sum_{\forall \hat{\mathbf{x}} \in \mathbb{S}} w_d(z) \cos^{-1}\left(\frac{\mathbf{N}_{\text{q}} \cdot \mathbf{N}_{\text{f}}}{|\mathbf{N}_{\text{q}}||\mathbf{N}_{\text{f}}|}\right), \tag{3}$$

where $w_d(z)$ is the weight considering the distortion rate of the equirectangular projection. This distortion weight $w_d(z)$, is calculated along the latitude z of the sphere, *i.e.*, $w_d(z) = \sqrt{1 - z^2}$, which effectively limits the influence of the top and bottom regions on the result. A visualization of the distortion weight is shown in Fig. 5.

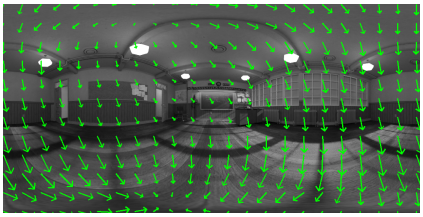
Minimizing the $\boldsymbol{\Omega}$ enables the network to estimate the 3 DoF rotation parameters (α, β, γ) and 2 DoF parameters (θ, ϕ) of the translational direction, *i.e.*, the 5 DoF motion of the spherical camera. This angular minimization can be achieved by using a geometric constraint to ensure that the large circle of the derotated optical flow corresponds to the



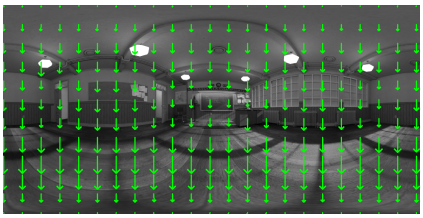
(a) Definition (b) Distortion Weight

FIGURE 5. Distortion weight. The equirectangular images have different distortions along the latitude of the sphere z. Therefore, the distortion weight $w_d(z)$ with values in the range of 0 to 1, is adopted. This enables the network to consider the varying contribution of the distortion.

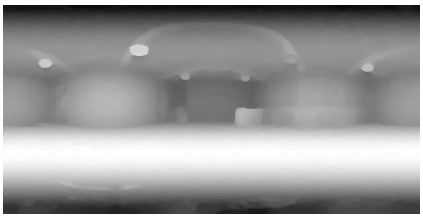
(a) Raw Frame: equirectangular projection



(b) Spherical Optical Flow: translation + rotation



(c) Derotated Optical Flow: translation only (+z direction camera moving)



(d) Disparity Map: the magnitude of (c)

FIGURE 6. Spherical optical flow derotation. Spherical optical flow (a) estimated from two raw frames, (b) consists of the mixture of the rotational and translational components. This can be derotated on a sphere, decoupling each component. After the derotation, (c) only the translational component remains, (d) whose magnitude corresponds to the disparity of the two frames. This derotation operation is unique to the spherical camera, which can acquire all-round optical flow on the sphere.

epipolar circle. Here, the direction of the translation is estimated by using this minimization. Consequently, an additional condition is required to acquire the translation scale, to obtain the full 6 DoF motion. This can be realized by triangulating the depth map by fixing the translation scale from I_t to I_{t+1} as 1, which is explained in the next section.

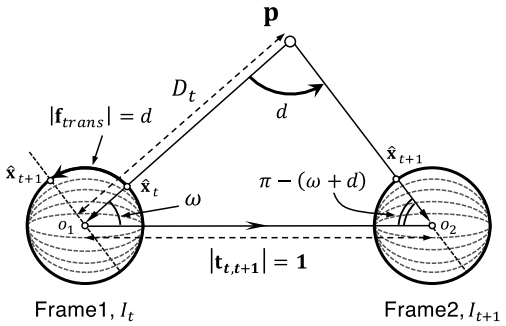


FIGURE 7. Triangulation. The disparity, which corresponds to the magnitude of the derotated (translation only) optical flow d, is converted into the depth D_t through the triangulation of the measured point p. Here, the translation between two frames $|\mathbf{t}_{t,t+1}|$ is fixed to 1, without loss of generality.

B. TRIANGULATION

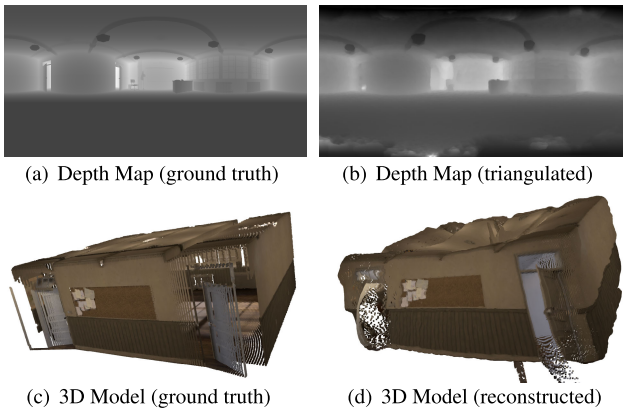
The triangulation used for calculating the depth map is explained as follows. The first two consecutive frames I_t, I_{t+1} of the equirectangular projection are used for obtaining the optical flow $\mathbf{f}$, which represents a mixture of the translational and rotational components. Later, the derotated optical flow and the translation direction are acquired by minimizing the 5 DoF epipolar angular error as explained in Section III-A. Furthermore, the disparity, which corresponds to the magnitude of the derotated optical flow $|\mathbf{f}_{\text{trans}}|$, can be converted into the depth map D_t by using the per-pixel triangulation (Fig. 7):

$$D_t = |\mathbf{t}_{t,t+1}| \times \frac{\sin(\omega + d)}{\sin(d)}, \tag{4}$$

where $|\mathbf{t}_{t,t+1}|$ is the translation of the two frames $I_t \rightarrow I_{t+1}$, ω is the angle between the translational, *i.e.*, epipolar, direction and the spherical unit vectors $\hat{\mathbf{x}}$, and d is the disparity which is equal to $|\mathbf{f}_{\text{trans}}|$ on the unit sphere. In this case, as explained earlier, $|\mathbf{t}_{t,t+1}|$ is fixed to 1, without loss of generality. To emphasize, all the variables required for this triangulation can be obtained from the minimization conducted in Section III-A. The translation scale is fixed to allow a relative scale to be estimated from the next frame, which is explained in Section III-C.

The spherical image is stretched onto the planar equirectangular image. Consequently, the generated depth map is distorted, which roughens the synthetic image, especially the top and bottom areas, as shown in Fig. 8. To relieve this distortion in training, Gaussian filter ($\sigma = 2$) with a size of 7×7 is used as a smoothing operation.

An example of the triangulated depth map and the per-pixel reconstructed 3D model with ground-truth are shown in Fig. 8. The depth map is triangulated from the magnitude of the translational optical flow, *i.e.*, disparity. To confirm the validity of the generated depth map, it is used to reconstruct a 3D model. As shown in Fig. 8(d), the ceiling and floor regions of the reconstructed 3D model appear distorted. This is due to the distortion of the spherical image, which is apparent in the upper and lower parts of the stretched equirectangular image, *i.e.*, the distorted depth



(a) Depth Map (ground truth) (b) Depth Map (triangulated)

(c) 3D Model (ground truth) (d) 3D Model (reconstructed)

FIGURE 8. Depth map and 3D model. The ground-truth and triangulated depth map (a), (b), and reconstructed 3D model (c), (d). In (a), (b), the colors black and white represent the closer and farther regions, respectively.

map. This distortion leads to incorrect estimation, which can be resolved by adopting the distortion weight explained in Section III-A.

C. PHOTOMETRIC REPROJECTION ERROR

Once the depth map is obtained, the third frame I_{t+2} can be reprojected by using spherical warping. This generates a synthetic image $\hat{I}_{t+2}$ of the third frame I_{t+2} using a transformation matrix $\mathbf{T}_{t,t+2}$, which is a combination of $\mathbf{R}_{t,t+2}$ and $\mathbf{t}_{t,t+2}$. The synthetic image is generated using a pixel-wise binary interpolation to fill the gaps, which is similar to [33]. The reprojected third frame $\hat{I}_{t+2}$ and the target frame I_{t+2} are required to be the same. Therefore, the photometric reprojection error $\mathbf{\Delta}$ can be defined in all pixels p. Moreover, to prevent the loss minimization from getting stuck in the local minima, $\mathbf{\Delta}$ consists of the aggregation of multi-scale resolution of I_{t+2} and $\hat{I}_{t+2}$. The resolutions of I_{t+2} and $\hat{I}_{t+2}$ are downsampled as $1/2^s$. Finally, the full 6 DoF motion of $I_t \rightarrow I_{t+2}$ is estimated by minimizing $\mathbf{\Delta}$, as follows:

$$\mathbf{\Delta} = \sum_s \sum_p w_d(z) \| I_{t+2}(p) - \hat{I}_{t+2}(p) \|_1, \tag{5}$$

where s indexes a downsampling scale of the multi-scale reprojected image, and $w_d(z)$ is the same distortion weight that is described in Section III-A. The effectiveness of the multi-scale reprojection and the distortion weight $w_d(z)$ were demonstrated by conducting experiments and the results are presented in Section V.

To summarize, the camera position of the first frame I_t is set to be the origin and the 5 DoF motion of $I_t \rightarrow I_{t+1}$ is estimated by minimizing the epipolar angular error $\mathbf{\Omega}$ (Section III-A) with the scale of translation set as 1, without the loss of generality (Section III-B). Later, the full 6 DoF motion of $I_t \rightarrow I_{t+2}$ is estimated by minimizing the photometric reprojection error $\mathbf{\Delta}$ proposed in this section. The entire process does not require the use of explicit labels, thus making it possible for the network to learn in a self-supervised manner.

IV. SELF-SUPERVISED LEARNING NETWORK

Our SelfSphNet structure is summarized in Fig. 9. Three consecutive frames are entered into the network as inputs and two losses are minimized to optimize the two camera motion parameters as outputs. The first and second image frames I_t and I_{t+1}, are used to calculate dense optical flow using the EpicFlow [21] method. Moreover, the first and third image frames I_t and I_{t+2} are entered into the first two convolution layers separately, and are then concatenated in the third convolution layer. Two parallel streams (*Stream A* and *B*) of CNNs are adopted to extract the features of each input, namely, the optical flow and stacked image frames. At the end of each stream, these features are flattened by global average pooling [41]. The final outputs of the fully connected layers for the regression consist of two loss minimization parts. First, the 3 DoF rotation $\mathbf{R}_{t,t+1}$ and the 2 DoF translational direction $\hat{\mathbf{t}}_{t,t+1}$ of $I_t \rightarrow I_{t+1}$ are estimated using the loss function $\mathcal{L}_{\text{epi}}$, which minimizes the epipolar angular error $\mathbf{\Omega}(I_t, I_{t+1})$ (Section III-A) in the entire training data i:

$$\mathcal{L}_{\text{epi}} = \sum_i \mathbf{\Omega}(I_t, I_{t+1}). \tag{6}$$

Next, the depth map is calculated via triangulation from the disparity map, which corresponds to the magnitude of the derotated optical flow $\mathbf{f}_{\text{trans}}$, of $I_t \rightarrow I_{t+1}$, as explained earlier. To estimate the full 6 DoF motion of $I_t \rightarrow I_{t+2}$, *i.e.*, the 3 DoF rotation $\mathbf{R}_{t,t+2}$ and the 3 DoF translation $\mathbf{t}_{t,t+2}$, the photometric reprojection error $\mathbf{\Delta}(I_t, I_{t+2})$, explained in Section III-C, is adopted as the loss function $\mathcal{L}_{\text{rep}}$ between the target and the synthetic frames in the entire training data i:

$$\mathcal{L}_{\text{rep}} = \sum_i \mathbf{\Delta}(I_t, I_{t+2}). \tag{7}$$

The total loss function $\mathcal{L}_{\text{self}}$ of our SelfSphNet consists of the combination of the 5 DoF epipolar angular loss $\mathcal{L}_{\text{epi}}$ and full 6 DoF photometric reprojection loss $\mathcal{L}_{\text{rep}}$, which are jointly minimized:

$$\mathcal{L}_{\text{self}} = \lambda_{\text{epi}} \mathcal{L}_{\text{epi}} + \lambda_{\text{rep}} \mathcal{L}_{\text{rep}}, \tag{8}$$

where λ_{epi} and λ_{rep} are scale factors to balance the weights of the two losses. To emphasize, this minimization is carried out without any labeled training data and requires only the consecutive raw frames.

V. EXPERIMENTAL RESULTS

To verify the performance of our proposed SelfSphNet, which estimates the motion of a spherical camera, we conducted experiments to compare the motion and trajectory estimation with a fully supervised learning network, as well as with another self-supervised motion estimation method named SfMLearner [33]. Another experiment was conducted as additional evaluation to ensure that transfer learning could be used to apply our SelfSphNet in an entirely new environment.

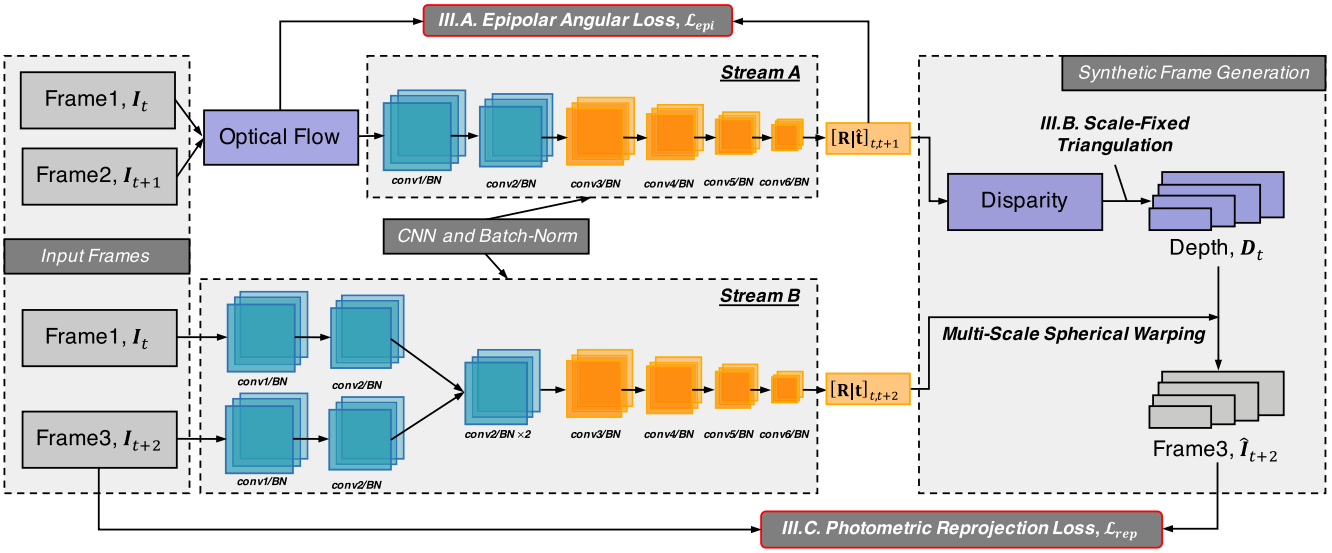


FIGURE 9. SelfSphNet structure. The optical flow generated from two input frames I_t, I_{t+1}, and the raw frames of I_t, I_{t+2} enter into two separate CNNs for feature extraction, *Stream A* and *B* respectively. The final outputs regress the 5 DoF motion of $I_t \rightarrow I_{t+1}$ and the full 6 DoF motion of $I_t \rightarrow I_{t+2}$, by the loss minimization which consists of the epipolar angular error and photometric reprojection error. The depth map is generated by the triangulation of the disparity map for synthesizing the third frame I_{t+2}. Furthermore, the spherical warping is conducted using the 6 DoF motion of $I_t \rightarrow I_{t+2}$ and the triangulated depth map.

A. TRAINING SPHERICAL CAMERA DATASET

Existing studies on self-supervised motion estimation [31], [33] have shown an almost two-dimensional estimation using the traditional KITTI dataset [42] for self-driving vehicles. Consequently, a fair evaluation in the case of an arbitrarily moving camera, which is our problem setting, would be complicated. Furthermore, a spherical camera, which captures 360° images, cannot be tested equivalently with networks that were trained by images from a perspective camera. Consequently, we manually constructed the spherical camera dataset using the rendering software Blender [43], the camera models of which have no direct influence on sensor errors or motion blurs. Details of the composition of this dataset are provided in Section V-C. Real spherical images, with ground truths (for evaluation) of the camera motion, were captured from a classroom scene,[1] as shown in Fig. 6 and 8. All the obtained images were projected onto a planar equirectangular image with a resolution of 200×100 pixels. Using our dataset, all networks were trained end-to-end for a fair comparison. In [33], the image warping part of the photometric reprojection loss was changed to accommodate spherical images because it originally considered perspective images.

B. COMPARISON WITH A FULLY SUPERVISED LEARNING

For comparison with our SelfSphNet, an additional experiment was conducted using a fully supervised learning method whose network baseline was similar to that of SelfSphNet (Fig. 9). This fully supervised learning network was provided with the ground-truth labels as supervision. Therefore, the Euclidean distances between the ground-truth and estimated values were minimized as a loss function, which is similar to [12]. This was realized by deleting the part of our SelfSphNet that generates the synthetic frame (Fig. 9).

Fully supervised learning with labeled training data was conducted using the combination of the following two L_2 loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$, between the ground truths and the estimated values of two frame pairs, I_t, I_{t+1} and I_t, I_{t+2}, respectively:

$$\mathcal{L}_1 = \left\| \hat{\mathbf{q}}_{t,t+1} - \frac{\mathbf{q}_{t,t+1}}{\|\mathbf{q}_{t,t+1}\|} \right\|_2 + \lambda_1 \left\| \hat{\mathbf{t}}_{t,t+1} - \mathbf{t}_{t,t+1} \right\|_2, \quad (9)$$

$$\mathcal{L}_2 = \left\| \hat{\mathbf{q}}_{t,t+2} - \frac{\mathbf{q}_{t,t+2}}{\|\mathbf{q}_{t,t+2}\|} \right\|_2 + \lambda_2 \left\| \hat{\mathbf{t}}_{t,t+2} - \mathbf{t}_{t,t+2} \right\|_2, \quad (10)$$

where $\hat{\mathbf{q}}_{t,t+1}$, $\hat{\mathbf{q}}_{t,t+2}$ and $\mathbf{q}_{t,t+1}$, $\mathbf{q}_{t,t+2}$ were the ground-truth and the estimated value, respectively, of the quaternion rotation; $\hat{\mathbf{t}}_{t,t+1}$, $\hat{\mathbf{t}}_{t,t+2}$ and $\mathbf{t}_{t,t+1}$, $\mathbf{t}_{t,t+2}$ were the ground-truth and the estimated value of the metric translation; and λ_1, λ_2 (0.1, 0.1 in this paper, similar to those in [39], [44]) were the scale factors used to balance the two loss functions in $\mathcal{L}_1$, $\mathcal{L}_2$, respectively. The total loss function $\mathcal{L}_{\text{sup}}$ for the fully supervised learning was a combination of the two losses $\mathcal{L}_1$, $\mathcal{L}_2$ as given below:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_1 + \mu \mathcal{L}_2, \quad (11)$$

where μ is a scale factor used to balance the two losses $\mathcal{L}_1$, $\mathcal{L}_2$, and was set to 1.0 in this case.

C. DATASET AND NETWORK COMPOSITION

This section details the spherical camera dataset and the structure of our SelfSphNet are detailed. To assume arbitrary movement of the spherical camera and to enable computation

[1] Available under CC0 license in http://www.blender.org.

of the optical flow, the rotation for each angle-axis (roll, pitch, and yaw) was set within $-5°$ to $5°$, and the translation for each axis (x, y, and z) was set within -0.1 to 0.1 meters, randomly, between all frames. The quantity of frames in the training, validation, and test datasets amounted to 18,148, 2,202, and 1,647 frames as an equirectangular projection, respectively. For the entire collection of data, the ground truths of the camera motion, *i.e.*, rotation and translation, were captured only for the quantitative evaluation. The dense optical flow generated from two consecutive frames was decomposed into horizontal and vertical vector components, and was then stored in two separate channels.

Our proposed SelfSphNet adopted two kinds of loss functions, which required minimizing them into two main streams, *Stream A* and *Stream B*, and consequently, we adopted two CNNs as feature extractors. The network blocks of *Stream A* consisted of conv1/BN$_{[16]}$, conv2/BN$_{[32]}$, conv3/BN$_{[64]}$, conv4/BN$_{[128]}$, conv5/BN$_{[256]}$, and conv6/BN$_{[256]}$. The notation of conv/BN$_{[c]}$ is the combination of a convolution layer with c filters of size 7/5/3/3/3/3 with stride 2×2, and a batch normalization (denoted as BN) [45] layer before a nonlinear ReLU [46] activation layer. We found that batch normalization exhibited stable convergence of the multiple losses. The composition of *Stream B* was similar to that of *Stream A*, except for the concatenation after two conv/BN layers. After feature extraction, each last convolution layer in both streams was flattened using global average pooling [41]. The outputs of the two streams were two motion parameters, $[\mathbf{R}_{t,t+1}, \hat{\mathbf{t}}_{t,t+1}]$ and $[\mathbf{R}_{t,t+2}, \mathbf{t}_{t,t+2}]$, which were regressed by $\mathcal{L}_{\text{self}}$. The outputs $\mathbf{R}_{t,t+1}$ and $\mathbf{R}_{t,t+2}$ were normalized as 1, to convert them into a quaternion representation, which is numerically stable compared to the Euler-angle configuration. To give the network a good starting point for the learning, the initial value of the quaternion $[q_w, q_x, q_y, q_z]^T$ was set to $[1, 0, 0, 0]^T$, which implies zero rotation. The direction of the translation $\hat{\mathbf{t}}_{t,t+1}$ was also normalized to consider a unit sphere.

D. TRAINING DETAILS

The learning process was conducted for 100 epochs using the Adam optimizer [47]. In the beginning, the learning rate was 0.0002, which then decreased to 0.0001 after 60 epochs, and the batch size was fixed to 4 in the entire training process. The scale factors of the two loss functions were determined by specifying an initial set of λ_{epi}, λ_{rep} as 0.6 and 1.0, respectively. The minimization of $\mathcal{L}_{\text{rep}}$ was largely dependent on the depth map, which was estimated from the minimization of $\mathcal{L}_{\text{epi}}$, therefore, λ_{epi} was set to be gradually reduced to 0.15 during 20 epochs. We further confirmed that the total loss $\mathcal{L}_{\text{self}}$ converged more stably with this setting. This entire process was implemented on an NVIDIA GeForce RTX 2080 Ti (GPU) and an Intel Core i9-7900X (CPU). In terms of the computational time, the training including the optical flow calculation took approximately 5.3 hours and the testing took approximately 6 minutes (5 fps).

E. EVALUATION FOR MOTION ESTIMATION

A comparison with fully supervised learning and [33] was conducted on the same classroom dataset. The results of estimating the spherical camera motion were evaluated on the entire test data. The evaluation indices of the motion error in the N test data are as follows:

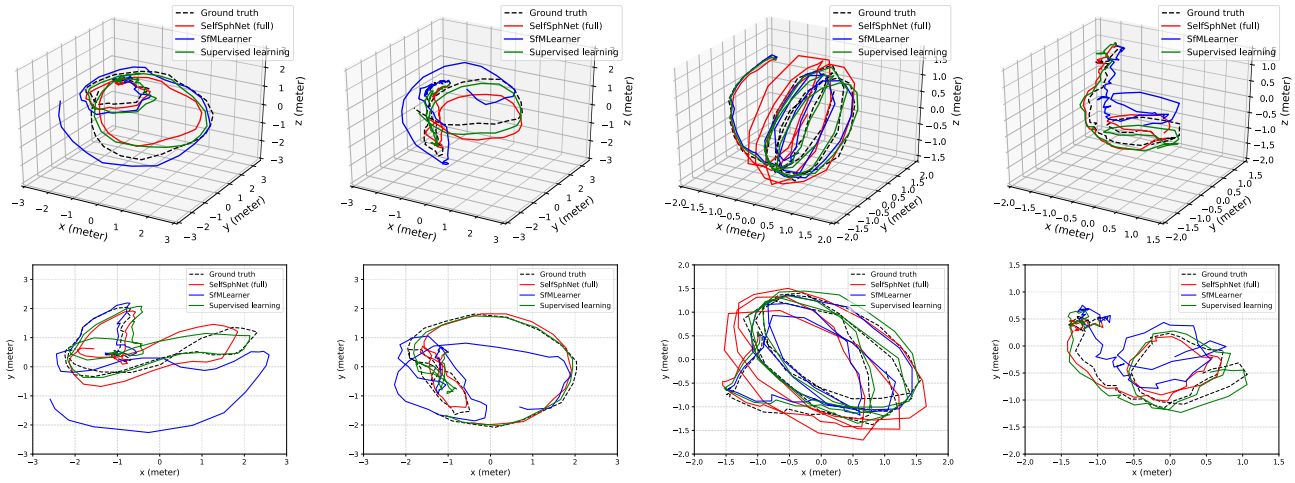
$$\text{Rotation Error} : \frac{1}{N}\sum_i 2\cos^{-1}(\hat{\mathbf{q}}_i \cdot \mathbf{q}_i), \tag{12}$$

$$\text{Translation Error} : \frac{1}{N}\sum_i \|\hat{\mathbf{t}}_i - \mathbf{t}_i\|_2. \tag{13}$$

The rotation errors between the ground-truth quaternion $\hat{\mathbf{q}}_i$ and the estimated quaternion $\mathbf{q}_i$ were evaluated as an average of the angular error (in degrees), in the angle-axis configuration. The translation errors between the ground-truth $\hat{\mathbf{t}}_i$ and the estimated $\mathbf{t}_i$ were evaluated as an average distance error (meters), along the x-, y-, and z-axes. To ensure a fair comparison, the estimated translation of our SelfSphNet was multiplied by the metric scale of $|\mathbf{t}_{t,t+1}|$, as our scale was fixed to 1, without loss of generality. The results of this camera trajectory estimation were also evaluated using the Absolute Trajectory Error (ATE) [48] index in various points (*Trajectory A, B, C,* and *D*), as shown in Fig. 10. In our dataset, the motion between frames was such that the optical flow could be calculated. Therefore, these frames can be regarded as keyframes. Considering this, 60 keyframes were used for evaluation in each trajectory.

All the estimation errors of the motion and trajectory are shown in Table 1. First, we confirmed that our SelfSphNet performed comparably with the fully supervised approach without the need for any labeled training data. Concretely, the errors in the rotation estimation were approximately 12.2% and the translation estimation approximately 29.3%. Next, the results showed that our SelfSphNet outperformed [33], in the rotation by approximately 31.6% and in the trajectory estimation by approximately 52.6%, 55.9%, 65.9%, and 59.7%, in *Trajectory A*, *B*, *C*, and *D*, respectively. However, its estimation of the translation was slightly lower by approximately 12.2%. Additionally, this proved that, as compared to the translation, the rotation is a more important factor in deciding the correct trajectory.

Moreover, to confirm the contribution of the distortion weight $w_d(z)$ and multi-scale reprojection in our SelfSphNet, ablation studies without these factors were also conducted, and the results are presented in Table 1. This shows an almost equal accuracy in the translation, whereas the rotation and trajectory results were less accurate compared with SelfSphNet (full). These results confirmed that the rotation was critically important to accurately estimate the trajectory. We confirmed that the distortion weight was effective to consider the varying contribution of the distortion in all pixels in minimizing the total-pixel error. We also confirmed that the various resolutions for composing the reprojection error enabled the network to achieve a coarse-to-fine loss minimization, which is robust in case of an image hole, such as textureless regions.



(a) Trajectory A (b) Trajectory B (c) Trajectory C (d) Trajectory D

FIGURE 10. Estimated trajectories. (Top) 3D trajectories estimated by our SelfSphNet (full), the fully supervised learning method, the previous SfMLearner [33], and the ground truth. (Bottom) 2D trajectories projected into the XY-plane.

TABLE 1. Ablation study and quantitative comparison. The entire end-to-end training was conducted with our spherical image dataset without pre-trained weights. To ensure a fair comparison, the image warping part of [33] was modified to consider the spherical projection. The fully supervised learning with our SelfSphNet baseline was also carried out with the labeled dataset. Except for the fully supervised learning, all networks were trained with the unlabeled dataset.

Network Architecture	Average Motion Error		Absolute Trajectory Error (m)			
	Rotation (deg.)	Translation (m)	*Trajectory A*	*Trajectory B*	*Trajectory C*	*Trajectory D*
SfMLearner [33]	0.610 ±0.349	**0.036 ±0.018**	0.646 ±0.280	0.998 ±0.358	0.990 ±0.421	0.444 ±0.142
SelfSphNet (w/o distortion weight)	0.463 ±0.381	0.044 ±0.025	0.815 ±0.357	**0.276 ±0.087**	1.140 ±0.614	0.509 ±0.261
SelfSphNet (w/o multi-scale)	0.433 ±0.403	0.042 ±0.025	0.399 ±0.278	0.474 ±0.230	0.603 ±0.259	0.535 ±0.332
SelfSphNet (full)	**0.417 ±0.379**	0.041 ±0.025	**0.306 ±0.155**	0.440 ±0.138	**0.338 ±0.121**	**0.179 ±0.085**
Supervised (SelfSphNet baseline)	0.366 ±0.353	0.029 ±0.025	0.484 ±0.352	0.325 ±0.212	0.413 ±0.159	0.175 ±0.065

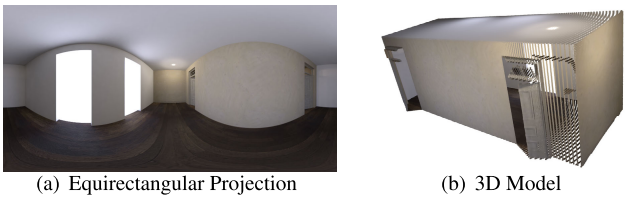
F. TRANSFER LEARNING IN NEW ENVIRONMENTS

To confirm the versatility of our SelfSphNet, we used our network to conduct transfer learning in entirely new environments: an indoor corridor scene and an outdoor urban scene (constructed by [7]), as shown in Fig. 11 and Fig. 13, respectively. As previously explained, the fully supervised learning approach cannot conduct transfer learning if the training data are not labeled, whereas our SelfSphNet can realize learning by capturing unlabeled frames. Specifically, our network was retrained for 30 epochs with a fixed learning rate of 0.0001 and a batch size of 64. In addition, all retraining was accomplished by setting the pre-trained parameters of the classroom scene as initial parameters. The retraining time was approximately 7 minutes. The quantities of the training frames were 5,271 and 4,991, in the corridor and the urban scenes, respectively. In addition, 896 and 332 frames were used for testing, respectively.

First, the estimation results in the corridor scene (Fig. 11) are provided in Table 2, with the estimated trajectory during 40 keyframes (Fig. 12). In Table 2, we confirmed that the transfer learning drastically increased the accuracy of the estimated translation and trajectory by approximately 50.0% and 48.4%, compared to the fully supervised learning approach, respectively, and by approximately 44.4% and 47.6%, compared to SelfSphNet (not fine-tuned), respectively. In Fig. 12, the estimated trajectory of the fully supervised learning and SelfSphNet (not fine-tuned) were largely quite different from the ground truth, especially in the latter part. They both moved in a similar wrong direction, which suggests that the pre-training was overfitted in the classroom scene. Meanwhile, the fine-tuned SelfSphNet showed a good fit in the new corridor scene with only a few minutes (7 minutes) of retraining.

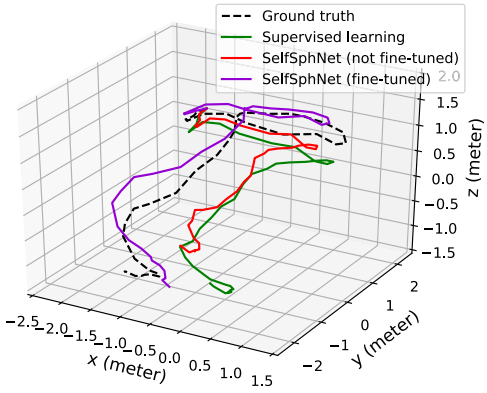
TABLE 2. Results in a corridor scene. Motion and trajectory estimation of the fully supervised learning, normal SelfSphNet, and the fine-tuned SelfSphNet.

Avg. Motion Error	Rot. (deg.)	Trans. (m)	ATE (m)
Supervised Learning	**0.60 ±0.33**	0.10 ±0.04	0.64 ±0.26
SelfSphNet (not fine-tuned)	0.74 ±0.36	0.09 ±0.04	0.63 ±0.23
SelfSphNet (fine-tuned)	0.62 ±0.31	**0.05 ±0.02**	**0.33 ±0.14**

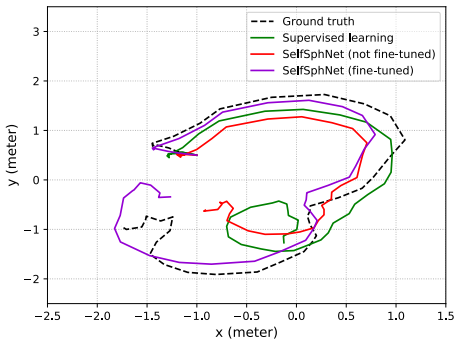


(a) Equirectangular Projection (b) 3D Model

FIGURE 11. Indoor corridor scene. A new, unknown scene.



(a) Corridor Scene Trajectory (3D)



(b) XY-plane (2D)

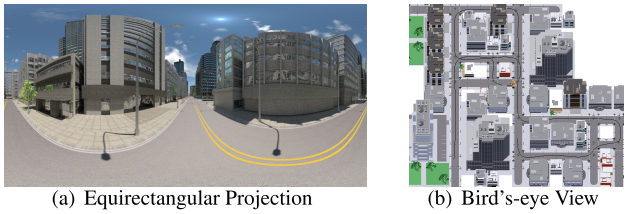
FIGURE 12. Fine-tuned trajectory in an indoor corridor scene. Fine-tuned SelfSphNet through transfer learning can estimate more accurately, compared to the fully supervised learning approach whose network cannot be retrained without the labeled training data.

The estimation results for the urban scene (Fig. 13) are presented in Table 3, with the estimated trajectory during 332 keyframes (Fig. 14). The results in Table 3 confirm that the rotation, translation, and trajectory estimation accuracy increased drastically by approximately 76.2%, 85.0%, and 87.4%, compared to the fully supervised learning method, respectively, and by approximately 66.1%, 88.3%, and 82.8%, compared to SelfSphNet (not fine-tuned), respectively. Fig. 14 shows that the trajectory estimated by our fine-tuned SelfSphNet (after transfer learning) is equivalent to the ground truth, whereas other trajectories were entirely incorrect.

All the results for the corridor and urban scenes indicate that our transfer learning method succeeded alleviating the overfitting in the trained classroom scene, thus optimizing the network using the newly collected unlabeled data from the indoor corridor scene and the outdoor urban scene. The

TABLE 3. Results in an urban scene. Motion and trajectory estimation of the fully supervised learning, normal SelfSphNet, and the fine-tuned SelfSphNet.

Avg. Motion Error	Rot. (deg.)	Trans. (m)	ATE (m)
Supervised Learning	0.84 ±0.49	0.60 ±0.03	25.9 ±8.94
SelfSphNet (not fine-tuned)	0.59 ±0.40	0.77 ±0.16	19.0 ±8.51
SelfSphNet (fine-tuned)	**0.20 ±0.08**	**0.09 ±0.06**	**3.27 ±1.25**



(a) Equirectangular Projection (b) Bird's-eye View

FIGURE 13. Outdoor urban scene. A new, unknown scene.

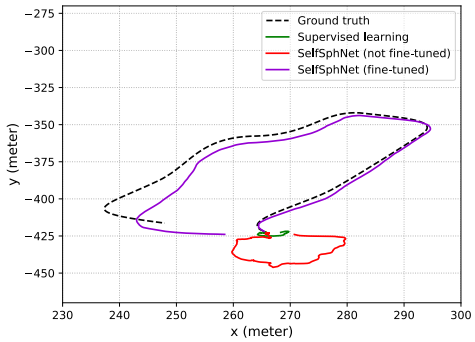


FIGURE 14. Fine-tuned trajectory in an outdoor urban scene. Fine-tuned SelfSphNet through a transfer learning can estimate more accurately, compared to the fully supervised learning approach whose network cannot be retrained without the labeled training data.

translation depends on the structure of each scene because the depth information varies across different scenes. Therefore, the translational optical flow differs and the estimation may fail because of the differences among these structures. Our transfer learning method overcame these problems by effectively filling the gap left by this difference and improved the accuracy of the trajectory estimation. This means our self-supervised learning network (SelfSphNet), which does not require any labeled training data, can learn the structure of the new scenes efficiently.

VI. DISCUSSION

Here, we discuss the experimental results, optical flow-based depth generation, and transfer learning in new scenes.

A. OPTICAL FLOW-BASED DEPTH GENERATION

The main purpose of this research was to propose a self-supervised learning network to estimate the motion of a spherical camera without using labeled training data. Estimation of the camera motion required the depth map to be

accurately estimated. Many previous studies have attempted to estimate the depth map from a single raw image, which was often affected by overfitting caused by RGB pixel intensities. Owing to overfitting in the trained environment, estimation in the new environment may be inaccurate, which implies that the generality of the learning is not guaranteed. To avoid this problem, our approach adopted the optical flow, which represents the pixel movement between two consecutive frames, and which is independent from the RGB pixel intensities. In addition, the optical flow generated from spherical images can be derotated and can decouple the rotational and translational components separately. As a result, the depth map could be generated using the derotated optical flow. The results in Table 1 and Fig. 10 show that our optical flow-based approach outperformed the single image-based approach [33] in terms of trajectory estimation. However, one drawback of this approach is that the motion between frames has to be sufficiently large to enable the optical flow to be accurately calculated. To resolve this, we ensured that the motion occurred within the required ranges during data collection. In addition, it is necessary to adopt a keyframe selection method as a pre-processing step when using datasets captured in real environments.

B. TRANSFER LEARNING IN NEW SCENES

Self-supervised learning has the advantage that it can train the network using unlabeled training data, whereas fully supervised learning requires labeled data. This indicates that the network can be fine-tuned by using the newly collected data captured in untrained environments via self-supervised learning. In the transfer learning experiment, our SelfSphNet that was pre-trained on the classroom scene, was retrained by using new datasets captured in the new environment. The experiment demonstrated that the fine-tuned SelfSphNet produced superior results compared with the fully supervised learning approach. Considering real applications, it is essential to conduct rapid retraining (approximately 7 minutes) by using the unlabeled data. Our transfer learning approach could realize this because it did not require a time-consuming labeling work of the retraining data.

VII. CONCLUSION

In this research, we proposed a self-supervised learning network (SelfSphNet) to estimate the motion of an arbitrarily moving spherical camera. Using the unique properties of the spherical camera, two loss minimizations were conducted to estimate the camera motion without labeled training data. The epipolar angular loss made it possible to estimate 5 DoF motion parameters between the first two consecutive frames. Subsequently, the disparity map, which corresponds to the magnitude of the derotated (translational) spherical optical flow, was triangulated to fix the translation scale from the next frames. The triangulated depth map was used to synthesize the third frame and the photometric reprojection loss was minimized to estimate the full 6 DoF camera motion. The novelty of our idea is that we designed and structured the self-supervised learning network to estimate the 6 DoF camera motion using the two loss functions. These loss functions were unique to the spherical camera because the optical flow was derotated to decouple the translational and rotational components in the 5 DoF epipolar angular loss. In addition, spherical warping was employed to generate the synthetic images in the photometric reprojection loss. This process obviates the need to use labeled training data for our network; instead, a collection of simple raw images without labels is sufficient.

Experiments demonstrated that our SelfSphNet outperformed a previous self-supervised method, SfMLearner, with respect to camera trajectory estimation. To ensure the effectiveness of our self-supervised method, and obtain a lower bound for the estimation error, fully supervised learning adopting the same baseline as SelfSphNet was also conducted using labeled training data. As a result, we confirmed that the results obtained with fully supervised learning were comparable to those of SelfSphNet.

To confirm the versatility of our SelfSphNet, we conducted transfer learning of the network in entirely new environments: an indoor corridor scene and an outdoor urban scene. This transfer learning experiment showed our SelfSphNet can learn newly collected data without time-consuming labeling work. As a result, the accuracy with the transfer learning was much higher than that obtained with fully supervised learning.

The frame-to-frame motion error was allowed to accumulate by including a large displacement in the estimated trajectory. Therefore, the use of an optimization method such as loop closure is required to recognize the points that have already been visited in the entire trajectory. Furthermore, training with additional frames should be explored. These tasks remain as future works.

REFERENCES

[1] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun./Jul. 2004, pp. 652–659.

[2] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numer.*, vol. 60, pp. 305–364, May 2017.

[3] A. Pagani and D. Stricker, "Structure from motion using full spherical panoramic cameras," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 375–382.

[4] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "A survey of variational and CNN-based optical flow techniques," *Signal Process., Image Commun.*, vol. 72, pp. 9–24, Mar. 2019.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[6] C. Gamallo, M. Mucientes, and C. V. Regueiro, "Omnidirectional visual SLAM under severe occlusions," *Robot. Auto. Syst.*, vol. 65, pp. 76–87, Mar. 2015.

[7] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 801–808.

[8] H. Taira, Y. Inoue, A. Torii, and M. Okutomi, "Robust feature matching for distorted projection by spherical cameras," *IPSJ Trans. Comput. Vis. Appl.*, vol. 7, pp. 84–88, Jul. 2015.

[9] S. Pathak, A. Moro, A. Yamashita, and H. Asama, "Dense 3D reconstruction from two spherical images via optical flow-based equirectangular epipolar rectification," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2016, pp. 140–145.

[10] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.
[11] H. Matsuki, L. von Stumberg, V. Usenko, J. Stuckler, and D. Cremers, "Omnidirectional DSO: Direct sparse odometry with fisheye cameras," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3693–3700, Oct. 2018.
[12] D. Kim, S. Pathak, A. Moro, R. Komatsu, A. Yamashita, and H. Asama, "E-CNN: Accurate spherical camera rotation estimation via uniformization of distorted optical flow fields," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2232–2236.
[13] J. Jiao, J. Jiao, Y. Mo, W. Liu, and Z. Deng, "MagicVO: An End-to-End hybrid CNN and bi-LSTM method for monocular visual odometry," *IEEE Access*, vol. 7, pp. 94118–94127, 2019.
[14] M. Lee and C. C. Fowlkes, "CeMNet: Self-supervised learning for accurate continuous ego-motion estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2019.
[15] X. Liu, J. V. D. Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to rank," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1862–1878, Aug. 2019.
[16] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised GANs via auxiliary rotation loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12154–12163.
[17] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018.
[18] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.
[19] S. Singh, A. Batra, G. Pang, L. Torresani, S. Basu, M. Paluri, and C. V. Jawahar, "Self-supervised feature learning for semantic segmentation of overhead imagery," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–12.
[20] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1385–1392.
[21] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1164–1172.
[22] F. Guo, Y. He, and L. Guan, "Deep camera pose regression using motion vectors," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4073–4077.
[23] S. Pathak, A. Moro, H. Fujii, A. Yamashita, and H. Asama, "Spherical video stabilization by estimating rotation from dense optical flow fields," *J. Robot. Mechatronics*, vol. 29, no. 3, pp. 566–579, Jun. 2017.
[24] A. Rituerto, L. Puig, and J. J. Guerrero, "Visual SLAM with an omnidirectional camera," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 348–351.
[25] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 141–148.
[26] J. Li, X. Wang, and S. Li, "Spherical-model-based SLAM on full-view images for indoor environments," *Appl. Sci.*, vol. 8, no. 11, p. 2268, Nov. 2018.
[27] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.
[28] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 616–625.
[29] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 2250–2257.
[30] Q. Liu, R. Li, H. Hu, and D. Gu, "Using unsupervised deep learning technique for monocular visual odometry," *IEEE Access*, vol. 7, pp. 18076–18088, 2019.
[31] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 3828–3838.
[32] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2019, pp. 7063–7072.
[33] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6612–6619.
[34] G. Iyer, J. K. Murthy, G. Gupta, K. M. Krishna, and L. Paull, "Geometric consistency for self-supervised End-to-End visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 380–388.
[35] F.-E. Wang, H.-N. Hu, H.-T. Cheng, J.-T. Lin, S.-T. Yang, M.-L. Shih, H.-K. Chu, and M. Sun, "Self-supervised learning of depth and camera motion from 360° videos," in *Proc. 14th Asian Conf. Comput. Vis. (ACCV)*, 2018, pp. 53–68.
[36] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360° imagery," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 529–539.
[37] P. Frossard and R. Khasanova, "Graph-based classification of omnidirectional images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 860–869.
[38] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018.
[39] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.
[40] J. Gluckman and S. K. Nayar, "Ego-motion and omnidirectional cameras," in *Proc. 6th Int. Conf. Comput. Vis. (ICCV)*, Jan. 1998, pp. 999–1005.
[41] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, 2013.
[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.
[43] *Blender.Org*. Accessed: Dec. 21, 2019. [Online]. Available: https://www.blender.org/
[44] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *Proc. 18th Int. Conf. Adv. Concepts Intell. Vis. Syst. (ACIVS)*, Nov. 2017, pp. 675–687.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.
[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[47] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[48] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

DABAE KIM (Student Member, IEEE) received the B.E. degree from the Department of Mechanical Engineering and Materials Science, Yokohama National University, Japan, in 2018. He is currently pursuing the master's degree with the Department of Precision Engineering, The University of Tokyo, Japan. His main research interests are deep learning, visual odometry, and 360° image processing using spherical cameras.

SARTHAK PATHAK (Member, IEEE) received the B.T. and M.T. degrees from the Department of Engineering Design, IIT Madras, India, in 2014, and the Ph.D. degree from the Department of Precision Engineering, The University of Tokyo, Japan, in 2017. Since 2018, he has been a Postdoctoral Research Fellow of the Japan Society for the Promotion of Science (JSPS) with the Department of Precision Engineering, The University of Tokyo. His main research interests are localization, 3-D reconstruction, and SLAM, especially using 360° spherical cameras.

ALESSANDRO MORO received the B.S. degree from the Department of Computer Science, University of Udine, Italy, in 2006, and the Ph.D. degree from the Department of Computer Software Engineering, University of Trieste, Italy, in 2011. He was a Visiting Research Fellow of Computer Vision of Chuo University, Japan, in 2011. He was a Research Engineer with Ritec Inc., in 2012. From 2012, he has been a Visiting Research Fellow of computer vision with The University of Tokyo. His research interests span computer and human vision, computer graphics, 3-D reconstruction, and machine learning. His main research interests are human and object recognition and machine learning for robotic application.

ATSUSHI YAMASHITA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the Department of Precision Engineering, The University of Tokyo, Japan, in 1996, 1998, and 2001, respectively. From 1998 to 2001, he was a Junior Research Associate with the Institute of Physical and Chemical Research (RIKEN). From 2001 to 2008, he was an Assistant Professor with Shizuoka University. From 2006 to 2007, he was a Visiting Associate with the California Institute of Technology. From 2008 to 2011, he was an Associate Professor with Shizuoka University. Since 2011, he has been an Associate Professor with the Department of Precision Engineering, The University of Tokyo. His research interests include robot vision, image processing, and intelligent sensing for robots. He is a member of ACM, JSPE, RSJ, IEICE, JSME, IEEJ, IPSJ, ITE, and SICE.

HAJIME ASAMA (Fellow, IEEE) received the M.S. and Dr. Eng. degrees from The University of Tokyo, Japan, in 1984 and 1989, respectively. He worked at RIKEN, Japan, from 1986 to 2002, where he became a Professor of RACE, The University of Tokyo, in 2002. He has been a Professor of School of Engineering, The University of Tokyo, since 2009, and the Director of RACE, since 2019. He was an AdCom Member of the IEEE Robotics and Automation Society, from 2007 to 2009. He was a member of the Science Council of Japan, from 2014 to 2017. He has also been a Council Member, since 2017. He is a Fellow of JSME and RSJ. He received the SICE System Integration Division System Integration Award for Academic Achievement, in 2010 and the JSME Award (Technical Achievement), in 2018. He was the Vice-President of RSJ, from 2011 to 2012. He has been the President-Elect of IFAC, since 2017, and the President of the International Society for Intelligent Autonomous Systems, since 2014.