To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2020, 1–15

FULL PAPER

Self-Supervised Optical Flow Derotation Network for Rotation Estimation of a Spherical Camera

Dabae Kim, Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, Hajime Asama

Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo Ward, Tokyo, Japan;

(*v1.0 released November 2020*)

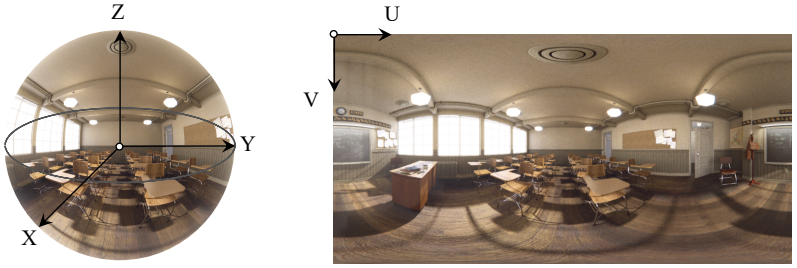
In this paper, we propose a self-supervised optical flow-based approach to learn the rotation of an arbitrarily moving spherical camera. Nowadays, deep learning has enabled efficient learning of camera rotation efficiently. However, most approaches are fully supervised and require large datasets with ground-truth labels of the rotation, and these labels are difficult to acquire. We attempt to solve this problem by using a derotation operation of the spherical optical flow on a unit sphere. This operation decouples the camera rotation from the mixture of translational and rotational components, removing the effect of 3D information for rotation estimation. Therefore, we integrate a derotation layer into a convolutional neural network for regressing the camera rotation. This layer can be adopted for only spherical cameras, which can capture all-round information, and thus enables the network to be learned the camera rotation without using labeled training datasets. We experimentally demonstrate that our approach achieves the comparable performance for the rotation estimation to that of a fully supervised approach, and that it outperforms a previously proposed approach. Moreover, transfer learning is conducted in new environments to confirm the benefit of the self-supervised learning.

Keywords: Self-supervised learning; rotation estimation; spherical camera; optical flow derotation; convolutional neural network

1. Introduction

Motion estimation of cameras is essential in robotic applications such as simultaneous localization and mapping (SLAM) [1] and structure from motion [2] as these applications require the movement of cameras. Recently, learning-based approaches have adopted for camera motion estimation using convolutional neural networks (CNNs) [3–5], and these approaches have performed equivalently or better than the conventional feature-based approaches. Among them, fully supervised learning approaches have been often utilized to regress the camera motion using raw images or optical flow fields as inputs. However, they require a large number of datasets with accurate labels, which are difficult to acquire. Many attempts have been made to capture such datasets using motion capture systems [6,7], GPS/IMU systems [8,9] and other sensors [10]. However, they often require precise sensor calibrations and collection/annotation expenses. Meanwhile, self-supervised learning approaches have recently emerged for scenarios in which it is difficult to acquire such labels, *e.g.* in the field of person re-identification [11], video hashing [12], and image classification [13]. These approaches offer the benefit that they do not require any labeled data, as they only utilize the collected data. For the camera motion estimation, many self-supervised approaches have been employed [14–17] and have shown estimation results comparable or better than fully supervised learning approaches.

Email: {kimdabae, pathak, moro, yamashita, asama}@robot.t.u-tokyo.ac.jp



(a) Spherical Projection (b) Equirectangular Projection

Figure 1. Spherical cameras can capture images as (a) spherical and (b) planar equirectangular projections owing to their 360° field-of-view.

Camera-based motion estimation has been explored in many robotic applications. Among various types of cameras, spherical cameras have attracted attention because of their large field-of-view. In addition, because of their 360° wide field-of-view (Figure 1), they are more beneficial than perspective cameras. This all-round view enables the estimation to be more robust against changes in environments, such as moving objects [18] or partially occluded regions [19]. Therefore, there has been an extensive body of research dedicated to the use of spherical cameras in tasks that involve motion estimation. These research have shed light on the advantages of having a 360° field-of-view [20–22]. To process spherical images in a plane, the captured images should be stretched to equirectangular projection (Figure 1(b)), similar to [23,24]. However, the equirectangular projection contains a distortion in the images, making the process difficult. To solve this problem, many studies have attempted to process spherical images in a spherical domain [25,26], and these studies have confirmed effective estimation.

Dense optical flow [27–29], which represents frame-to-frame pixels movement, has been commonly used to estimate the camera motion [30–32] or object motion [33,34]. In many previously conducted studies for learning-based camera motion estimation, raw images were often used to train the network. However, raw images make the training vulnerable to overfitting due to raw RGB pixel intensities. Even if the images were captured at the same location, the difference in RGB pixel intensities disturbs the training, and the results of the testing would be different. Meanwhile, optical flow can help make the training less vulnerable to raw RGB pixel intensities [32]. However, the optical flow for an arbitrarily moving camera still contains 3D structure information of environments, because of the parallax effect induced by camera translation. This results in multiple optical flow patterns with the same translation, making the learning difficult. Therefore, it is necessary to be able to distinguish between translation and rotation in optical flow fields.

As explained earlier, optical flow represents different patterns with the same motion due to camera translation, which is affected by different 3D structures of environments. Because of this difference, the network finds it difficult to learn camera rotation. To resolve this problem, we attempt to decouple the rotational optical flow from the mixture of translational and rotational components. This enables the network to avoid the overfitting problem, which is attributed to the translational optical flow. In this paper, we propose a learning-based method to estimate the rotation of a spherical camera. Especially, we design an optical flow derotation network to estimate the camera rotation in a self-supervised manner. This network explicitly rotates an input optical flow on a unit sphere and can be combined with CNNs. In addition, the network can regress the camera rotation without using any camera rotation label in the entire training data. After derotating the optical flow, a moment-based loss function, which is similar to that in [35], is used to estimate a 3 DoF camera rotation. We experimentally confirm that our derotation-based self-supervised approach obtains comparable results with that of the fully supervised approach and outperforms the results of a previously proposed self-supervised approach, *i.e.*, SfMLearner [36], in terms of rotation estimation. Furthermore, transfer learning is conducted in new environments to confirm the benefit of self-supervised learning, which can easily fine-tune

the network. The following are the contributions of this research.

- We propose a self-supervised learning network to estimate the spherical camera rotation without using labeled training data.
- A derotation layer is designed to decouple rotational and translational optical flow components, avoiding the overfitting problem.
- Transfer learning is conducted to confirm that the self-supervised approach can realize simple retraining in new environments.

2. Related work

Here, we introduce several research related to this study, fully supervised learning approaches and self-supervised learning approaches to estimate spherical camera rotation.

2.1 *Fully supervised approaches for camera rotation estimation*

Recently, deep learning-based approaches have been proposed to estimate a camera rotation [3–5,37]. In these approaches, the networks were often trained using a large number of images captured via cameras. While training the networks, the image features should be extracted via loss minimization, which enables the camera rotation to be regarded as a regression problem. The loss minimization often comprises the distance between ground truths and estimated values, such as L1-norm [38], L2-norm [4,5], and similarity metrics [37]. In camera rotation estimation, the loss functions are constructed using rotation representations such as Euler-angle [39], quaternion [32,40], and rotation matrix [41]. These approaches are referred to as fully supervised learning, as they require all the ground-truth labels of camera rotation. The ground-truth labels are critical to composing the above mentioned loss functions. Namely, fully supervised learning approaches can efficiently estimate the camera rotation using distance-based loss functions.

However, there is a drawback that these approaches cannot appropriately train the network unless the ground-truth labels of the camera rotation are provided. These labels are often acquired using motion capture systems [6,7] or GPS/IMU systems [8,9], as explained in Section 1. However, it is difficult to acquire accurate labels because of sensor errors or calibration errors. In addition, motion capture systems are limited in their application, as they can only be used in well-equipped spaces. GPS-based systems cannot function in situations wherein radio waves do not reach the systems because of obstacles such as buildings or bridges. Even if labels are acquired, overfitting, which lowers the estimation accuracy, may occur in new environments where the labels are not obtained. Therefore, a new training method to estimate camera rotation without using labels is required.

2.2 *Self-supervised approaches for camera rotation estimation*

In fully supervised learning approaches, labels of camera rotation are required, which are difficult to acquire. Meanwhile, self-supervised learning approaches have been proposed to estimate the camera rotation without using labels. These approaches provide pseudo-supervision signals to optimize networks, instead of direct supervision signals, which are formed using labels. For camera rotation estimation, the pseudo-supervision signals comprise certain loss functions. For example, [36,37] synthesize images to compose a photometric consistency loss and [15,17] constraint multiple transformation matrices geometrically. Especially, [15,38] attempt to estimate the rotation of a spherical camera by utilizing the benefit of its wide field-of-view. These approaches have shown a possibility to train the learning network without requiring direct labels of camera rotation.

However, since these methods require simultaneous estimation of camera rotation and transla-

tion, the translation may lower the accuracy of the rotation estimation. This can be a drawback when estimating only the rotation parameters in a mixture of the camera rotation and translation. In this paper, we propose a new method which can estimate only the rotation parameters without considering the effect of the camera translation.

3. Proposed method

Here, we explain the proposed self-supervised learning network to estimate the 3 DoF rotation of a spherical camera. First, we introduce a unique property of spherical optical flow; using the property, rotational and translational optical flow components are decoupled. Next, an optical flow derotation operation using the unique property is described with an optical flow moment-based loss function. Finally, we explain the proposed network, the combination of CNN and a derotation layer. This network does not require any explicit label of the training data.

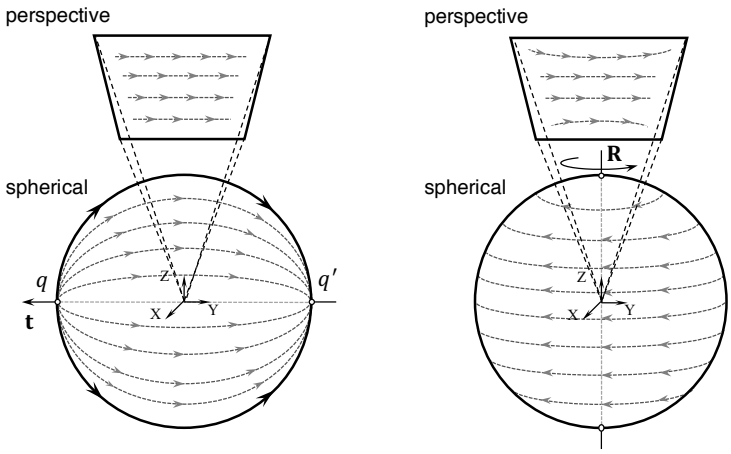
3.1 *Spherical optical flow*

An arbitrary motion of spherical cameras is a combination of translation and rotation. Furthermore, the optical flow of spherical images has a distinct capability of distinguishing between translation and rotation, as first elucidated in [42]. In a normal perspective camera, yawing to the counterclockwise direction and translating to the left are significantly similar to each other, making it difficult to distinguish between them, as shown in Figure 2. However, in a spherical camera, it is easy to figure out whether the camera is rotating or translating, as information can be accessed from all directions.

When spherical cameras are in the pure translational state, the optical flow of all the pixels diverge from the translation direction q and converge to its opposite pole q', as shown in Figure 2(a). Essentially, the optical flow vectors are directionally symmetric in the case of pure translation. However, in the case of the pure rotational state, the optical flow vectors move in circles, perpendicular to the rotational axis, as shown in Figure 2(b). These spherical optical flows can be computed using two planar equirectangular images, as shown in Figure 3. This distinguishing property of the spherical optical flow can be used to decouple the rotational and translational optical flow components. Specifically, a derotation operation can remove the effect of the rotational components from the mixture of the both the optical flow components on a sphere. In addition, this operation can be realized using a simple rotation matrix of all the axes. After derotating the optical flow, the translational components remain without any effect of the rotation.

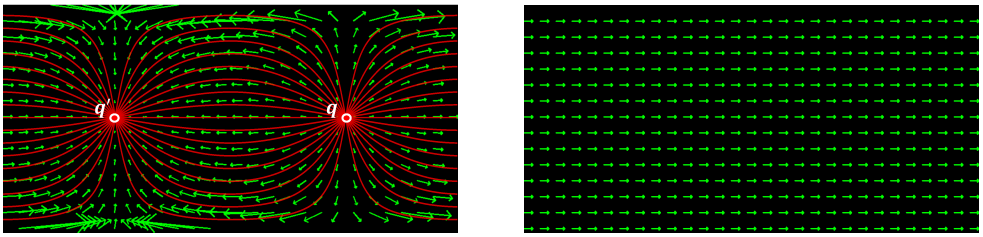
3.2 *Optical flow derotation*

The spherical optical flow can be derotated using an input optical flow data and output 3 DoF rotation parameters (α, β, γ) in the angle-axis. These parameters can be regressed without any prior information of the input optical flow data, *i.e.*, a label. Here, the label means a ground-truth of rotation parameters. This regression enables the network to learn camera rotation in a self-supervised manner, as the labels of the rotation parameters are not required in the derotation operation. To conduct this, we construct an optical flow derotation-based loss function to estimate the camera rotation without any prior information. This loss function utilizes a symmetric property of the optical flow on a unit sphere. A concrete explanation about the loss function will be described in the next Section 3.3. As a representation of the rotation parameter, we adopt a quaternion, $\mathbf{q} \in \mathbb{R}^4$, whose values are normalized as 1. Using the quaternion $\mathbf{q} = [q_\mathrm{w}, q_\mathrm{x}, q_\mathrm{y}, q_\mathrm{z}]$, a derotation matrix $\mathbf{R}_\mathrm{q}^T$ can be calculated on a unit sphere, as follows:



(a) Translation (b) Rotation

Figure 2. Optical flow on spherical and planar projections. (a) Spherical and perspective optical flow with the pure translational movement of $\mathbf{t}$ (from q to q') of a spherical camera, and (b) the pure rotational movement of $\mathbf{R}$ to the counterclockwise direction.



(a) Translation (b) Rotation

Figure 3. Optical flow computation results. The optical flow in the equirectangular projection with (a) the pure translation from q to q', and (b) the pure yawing rotation. The optical flow with the pure translation are seen that they are aligned along the red curved lines, indicating the epipolar curves.

$$\mathbf{R}_{\mathrm{q}}^{T} = \begin{bmatrix} 1-2q_{\mathrm{y}}^2-2q_{\mathrm{z}}^2 & 2q_{\mathrm{x}}q_{\mathrm{y}}+2q_w q_{\mathrm{z}} & 2q_{\mathrm{x}}q_{\mathrm{z}}-2q_w q_{\mathrm{y}} \\ 2q_{\mathrm{x}}q_{\mathrm{y}}-2q_w q_{\mathrm{z}} & 1-2q_{\mathrm{x}}^2-2q_{\mathrm{z}}^2 & 2q_{\mathrm{y}}q_{\mathrm{z}}+2q_w q_{\mathrm{x}} \\ 2q_{\mathrm{x}}q_{\mathrm{z}}+2q_w q_{\mathrm{y}} & 2q_{\mathrm{y}}q_{\mathrm{z}}-2q_w q_{\mathrm{x}} & 1-2q_{\mathrm{x}}^2-2q_{\mathrm{y}}^2 \end{bmatrix}. \tag{1}$$

Using the derotation matrix, images can be derotated directly, and the spherical optical flow can be re-calculated. However, the computational cost of this process is high when considering an iterative regression. To solve this high cost problem, we derotate the initial location of the optical flow $\hat{\mathbf{x}}_{\mathrm{i}} = [x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}}]^T$ rather than the images themselves. Namely, the vectors of the optical flow are derotated on a sphere. Therefore, the derotated location of the optical flow $\hat{\mathbf{x}}_{\mathrm{d}} = [x_{\mathrm{d}}, y_{\mathrm{d}}, z_{\mathrm{d}}]^T$, which is derived using the initial location $\hat{\mathbf{x}}_{\mathrm{i}}$, can be expressed as follows:

$$\hat{\mathbf{x}}_{\mathrm{d}} = \mathbf{R}_{\mathrm{q}}^{T}\, \hat{\mathbf{x}}_{\mathrm{i}}. \tag{2}$$

After conducting the derotation operation, the derotated optical flow vector $\mathbf{f}_{\mathrm{d}}$, which is derived using the initial optical flow vector $\mathbf{f}_{\mathrm{i}}$, can be expressed by subtracting $\hat{\mathbf{x}}_{\mathrm{d}}$ from the point at the end of the initial optical flow vector $\hat{\mathbf{x}}_{\mathrm{i}} + \mathbf{f}_{\mathrm{i}}$, as follows (Figure 4):

$$\mathbf{f}_{\mathrm{d}} = (\hat{\mathbf{x}}_{\mathrm{i}} + \mathbf{f}_{\mathrm{i}}) - \hat{\mathbf{x}}_{\mathrm{d}}. \tag{3}$$

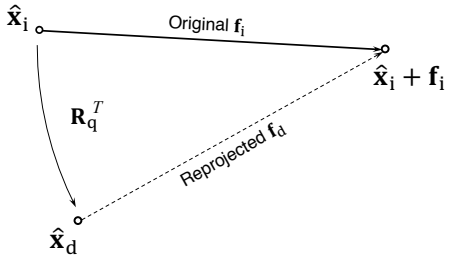


Figure 4. The initial optical flow vectors $\mathbf{f}_i$ are derotated using the rotation matrix of quaternion $\mathbf{R}_q^T$, and then the reprojected optical flow vectors $\mathbf{f}_d$ are obtained.

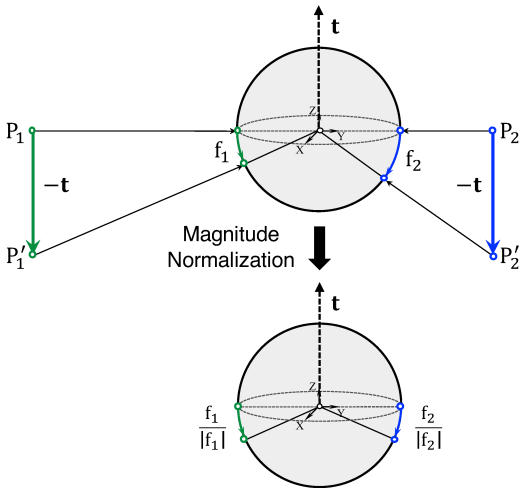


Figure 5. Magnitude normalization of the optical flow. In the pure camera translation of $\mathbf{t}$, polar opposite points on the sphere (P_1 and P_2) produce moments in the opposing directions. However, the magnitudes of the two optical flow projected on the sphere are different due to depth information in the opposite points. Accordingly, the optical flow $\mathbf{f}$ need to be normalized by their magnitudes $|\mathbf{f}|$. By normalizing the optical flow, the pure translational optical flow becomes depth-independent, keeping symmetry.

In the derotated optical flow vector $\mathbf{f}_d$, only the translational component remains. As explained in Section 1, the 3D structure information of the environments, which induce overfitting in the training, are included in the translational optical flow. Accordingly, the derotated translational optical flow $\mathbf{f}_d$ is normalized by its magnitude $|\mathbf{f}_d|$ for considering only its direction, as shown in Figure 5. In Figure 5, the opposite points P_1 and P_2 represent different magnitudes even if the camera translates with the same value of $\mathbf{t}$, as their distances from the camera center are different. This disturbs the symmetry of the optical flow in the pure translational state. Using this normalization, the derotated optical flow becomes symmetrical to the optical flow that is located in the opposite side.

To estimate the 3 DoF rotation parameters, we adopt an optical flow moment-based loss function, which comprises the normalized derotated optical flow and the location there of. Here, the above mentioned optical flow normalization enables the network to regress the 3 DoF rotation parameters by intrinsically ignoring the 3D information. The total optical flow moment $\mathbf{M}$ over every pixel $\hat{\mathbf{x}}_d$ on the sphere $\mathbb{S}$ can be calculated using the cross product of the pixel location and the normalized optical flow as follows:

$$\mathbf{M} = \sum_{\forall \hat{\mathbf{x}}_d \in \mathbb{S}} \left(\hat{\mathbf{x}}_d \times \frac{\mathbf{f}_d}{|\mathbf{f}_d|} \right). \tag{4}$$

The optical flow moment loss function can be minimized and can be used to estimate the 3 DoF rotation of the spherical camera without using any labels of the training data.

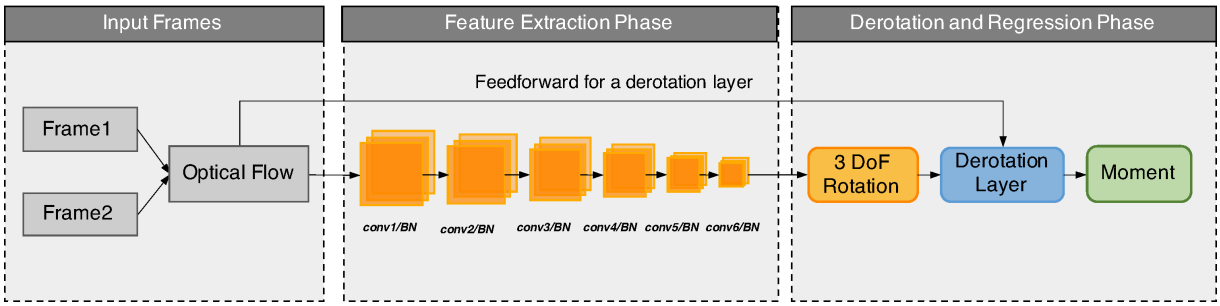


Figure 6. The proposed network. First, the unlabeled optical flow is generated from the two frame images with an arbitrary motion, and is entered into a feature extraction network of the convolutional neural network (CNN). Next, the optical flow is derotated on a sphere using the quaternion outputs, and the optical flow moment is calculated. Last, moment minimization regresses the rotation parameters in a single-pass fashion. The entire process can be conducted in a self-supervised manner without using any explicit labels of the camera rotation.

3.3 *Self-supervised learning network*

The proposed self-supervised learning network for rotation estimation of the spherical camera is summarized in Figure 6. The optical flow calculated using two frame images enters into the CNN for feature extraction, and the output consists of 3 DoF rotation parameters. Using the rotation parameters, the input optical flow is derotated in the derotation layer, as explained in Section 3.2. Here, the derotation layer conducts a geometric operation meant to calculate the moment. Subsequently, the derotated optical flow comprises the moment as a loss function $\mathcal{L}_{\text{self}}$, and the rotation parameters are regressed via loss minimization. The weight parameters of our network are updated through backpropagation started from the moment minimization in a single-pass fashion. The loss function $\mathcal{L}_{\text{self}}$ is minimized as an L2-norm $\|\cdot\|$ of the optical flow moment $\mathbf{M}_i$ in all the training data i, as follows:

$$\mathcal{L}_{\text{self}} = \sum_i \|\mathbf{M}_i\|_2. \tag{5}$$

At the end of our network, the output is normalized to consider a quaternion representation. In addition, the initial value of the quaternion is set to be $[1, 0, 0, 0]^T$, which is equivalent to zero rotation, to give the network a good starting point in training. This is also because two optical flow moments calculated at two symmetric points, whose angles from the translation direction are θ and $\theta - \pi$, respectively, comprise the same value of moment. The 3 DoF rotation parameters are estimated by minimizing $\mathcal{L}_{\text{self}}$. This estimation can be performed without using the labeled training data. The only requirements are the input spherical optical flow and 3 DoF rotation parameters to derotate the optical flow.

For the feature extraction network, the following blocks of the CNN were adopted: $\text{conv1/BN}_{[16]}$, $\text{conv2/BN}_{[32]}$, $\text{conv3/BN}_{[64]}$, $\text{conv4/BN}_{[128]}$, $\text{conv5/BN}_{[256]}$, $\text{conv6/BN}_{[256]}$, $\text{fc1}_{[512]}$, and $\text{fc2}_{[4]}$. The notation $\text{conv/BN}_{[c]}$ is the combination of convolution and batch normalization [43] layers with c filters of size 7/5/3/3/3/3 with stride 2×2 and a ReLU [44] activation layer. Finally, a fully connected layer $\text{fc}_{[n]}$ was used with n nodes. The final layer, $\text{fc2}_{[4]}$, was normalized in order to convert the output to the quaternion.

4. Experiments

To verify the effectiveness of our approach, several rotation estimation experiments were conducted including ablation studies. For quantitative comparison, a previously proposed SfMLearner [36] experiment was conducted using the same dataset without pre-trained weights. Furthermore, a fully supervised learning experiment, which used the labels of camera rotation, was also conducted. In the fully supervised learning experiment, we set two types of the input

data, namely, the optical flow and raw images, to demonstrate that the optical flow prevents overfitting, which the raw RGB pixel intensities often suffer from. First, we explain the manually collected dataset using a simulator. We then describe the training details and estimation evaluation. Finally, transfer learning is conducted in new environments to confirm the benefit of our self-supervised learning approach.

4.1 *Dataset composition*

The input optical flow was generated using raw images, which were rendered by Blender [45]. For the optical flow computation, we adopted MR-Flow [29]. All the raw images in the equirectangular projection were captured from a classroom scene[1], as shown in Fig 1. For the conditions of the dataset, the rotation angles between two frames were randomly chosen from within -5° to 5° in each axis (roll, pitch, and yaw). In addition, the translation along each axis (x, y, and z) was also randomly chosen from within -0.1m to 0.1m, to enable the optical flow computation. The quantity of training, validation, and test datasets amounted to 18,142, 2,196, and 1,641 frames, respectively. For the entire data, the ground truths of the rotation angles were recorded only for evaluation. The input data comprised two channels, namely, the horizontal and vertical components of the optical flow, with a resolution of 200×100 pixels.

4.2 *Training details*

The learning process was conducted with an Adam optimizer [46] for 100 epochs with the fixed learning rate of 0.0001 and batch size of 32. The entire computation was performed on an NVIDIA GeForce GTX 1080 (GPU) and an Intel Core i7-8700K (CPU). In terms of the computational time, the training and testing the included optical flow calculation took approximately 2.7 h and 6 min (5 fps), respectively.

4.3 *Rotation estimation evaluation*

We evaluated the proposed self-supervised learning approach by comparing it with a previously proposed method, namely, SfMLearner [36], and a fully supervised approach. They were trained using the same datasets, which were introduced in Section 4.1. All the training was conducted without any pre-trained weight. The fully supervised learning network, whose composition was the same as that of our network, adopted the Euclidean distance $\|\cdot\|$ between the ground-truth quaternion $\hat{\mathbf{q}}$ and normalized estimation value $\mathbf{q}/\|\mathbf{q}\|$ (similar to [5]), as the following loss function $\mathcal{L}_{\text{sup}}$:

$$\mathcal{L}_{\text{sup}} = \left\| \hat{\mathbf{q}} - \frac{\mathbf{q}}{\|\mathbf{q}\|} \right\|_2. \tag{6}$$

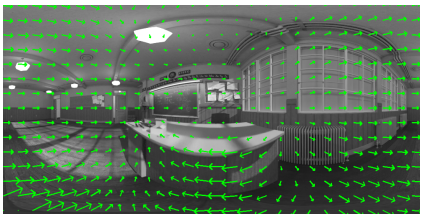
We found that the estimated quaternion becomes close to the ground-truth quaternion. Therefore, directly adopted the Euclidean distance loss to avoid unnecessary network optimization interference, which is also described in [32] and [40]. The estimation results of the spherical camera rotation were evaluated on the entire testing data. The evaluation metrics of the rotation error in all N testing data were the average rotation error (ARE) and median rotation error (MRE), as follows:

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

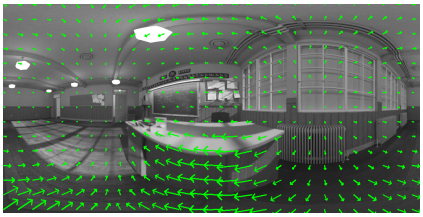
Table 1. Ablation studies and quantitative comparison.

Methods	**ARE (°)**	**MRE (°)**
SfMLearner [36]	0.610 ±0.349	0.544
Our (w/o BN)	0.390 ±0.284	0.349
Our (full)	**0.370 ±0.288**	**0.317**
Supervised (raw images)	0.529 ±0.395	0.469
Supervised (optical flow)	**0.324 ±0.255**	**0.286**

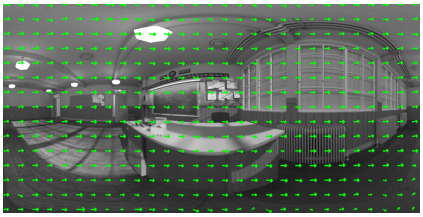
(a) Raw frame: equirectangular projection



(b) Input optical flow: translation + rotation



(c) Derotated optical flow: only translation



(d) Subtracted optical flow: (b)-(c), only rotation

Figure 7. The estimation example in the equirectangular projection. (a) The raw frame of the planar equirectangular projection, (b) input optical flow from the mixture of the translational and rotational camera movements, (c) translational optical flow calculated using the derotation operation with the estimated rotation parameters, and (d) rotational optical flow by subtracting of (c) from (b).

$$\mathrm{ARE} = \frac{1}{N}\sum_{i} 2\cos^{-1}(\hat{\mathbf{q}}_i \cdot \mathbf{q}_i), \tag{7}$$

$$\mathrm{MRE} = \mathit{Median}\big\{2\cos^{-1}(\hat{\mathbf{q}}_i \cdot \mathbf{q}_i)\big\}, \tag{8}$$

where ARE denotes an averaged angular error between the ground-truth quaternion $\hat{\mathbf{q}}_i$ and the estimated quaternion $\mathbf{q}_i$ of data i, and $\cdot$ depicts a scalar product. The term MRE denotes a median angular error between both the quaternions, and *Median* denotes a median value.

The ablation studies and quantitative comparison for rotation estimation are presented in Table 1, with the results of the proposed self-supervised learning approach, SfMLearner, and fully supervised learning approach. The evaluation metrics on the test data were set as ARE and MRE in the angle-axis configuration. From the results, it is evident that the proposed approach confirmed that ARE decreased by about 39.3% comparing to the SfMLearner and showed comparable performance with that of the fully supervised learning approach. In the ablation study regarding batch normalization, we confirmed that the batch normalization layer contributed the performance improvement, preventing from overfitting. This improvement implies that the batch normalization layer reduced internal covariate shift between the network parameters effectively.

From the fully supervised learning experiments, we also confirmed that the optical flow was a more robust training data than raw images, whose RGB pixel intensities induced the overfitting problem. In addition, ARE of rotation estimation trained using the optical flow was lower than

Table 2. Average rotation errors when using several optical flow calculation methods.

Optical flow methods	**EpicFlow** [28]	**DeepFlow** [27]	**MR-Flow** [29]
ARE (°)	0.501 ±0.349	0.436 ±0.306	**0.370 ±0.288**

(a) Urban scene (b) Corridor scene (c) Cafe scene

Figure 8. New environments for transfer learning. (a) The outdoor urban scene, (b) indoor corridor scene, and (c) indoor cafe scene. The transfer learning can be conducted by simple data collection without using the camera rotation labels.

Table 3. Data quantities in new environments.

Number of data	**Training data**	**Testing data**
Urban scene	4,990	332
Corridor scene	3,684	566
Cafe scene	4,588	771

that of the raw images by approximately 38.8%.

In Figure 7, we show an example of the rotational optical flow decoupled from the mixture of the translational and rotational optical flow components by using the derotation operation. We confirmed that our derotation network could decouple the rotational and translational optical flow components.

4.4 *Effect of optical flow estimation method*

We also conducted additional experiments to use other optical flow calculation methods, DeepFlow [27] and EpicFlow [28], and their results are described in Table 2. We confirmed MR-Flow [29] resulted the lowest ARE with the ground-truth rotation parameters. From the results, we found that our moment-based approach largely depends on the accuracy of the optical flow calculation. The accuracies are compared by the optical flow benchmark dataset named MPI Sintel Flow Dataset [47]. The MR-Flow method will be adopted from the following experiments.

4.5 *Transfer learning in new environments*

Our self-supervised learning approach has a benefit that it can train a network by using unlabeled training data. This indicates that only simple image collection is required without the time-consuming labeling task. Therefore, transfer learning can be quickly realized in new environments by our network, whereas a fully supervised learning approach cannot conduct transfer learning easily. We conducted transfer learning in entirely new environments, an outdoor urban scene [20] and indoor corridor[1], cafe scenes [48], as shown in Figure 8. In the transfer learning, pre-trained weights, which were trained in the classroom scene, were set as the initial weights of the network for giving a good starting point. All transfer learning in each scene was conducted for 60 epochs with a fixed learning rate of 0.0001 and batch size of 32. The quantities of the training and testing data in each scene are described in Table 3.

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

Table 4. Transfer learning (TL) results in various environments.

Methods	Urban scene		Corridor scene		Cafe scene	
	ARE (°)	MRE (°)	ARE (°)	MRE (°)	ARE (°)	MRE (°)
Supervised Learning	0.735 ±0.371	0.677	0.659 ±0.364	0.595	0.251 ±0.119	0.240
Our (before TL)	0.598 ±0.342	0.562	0.640 ±0.352	0.584	0.186 ±0.095	0.172
Our (after TL)	**0.142 ±0.065**	**0.133**	**0.507 ±0.280**	**0.449**	**0.174 ±0.109**	**0.156**

The estimation results in each scene are provided in Table 4. We confirmed that the transfer learning significantly decreased the rotation estimation error in all the environments. Compared with the fully supervised learning, ARE after the transfer learning decreased by approximately 80.7%, 23.1%, and 30.7% in the urban, corridor, and cafe scenes, respectively. In addition, compared with ARE before the transfer learning, ARE after the transfer learning were decreased by approximately 76.3%, 20.8%, and 6.5% in the urban, corridor, and cafe scenes, respectively.

The rotational optical flow shows the same pattern irrespective of the structure of environments, as the information acquired via the frame-to-frame camera rotation are the same. However, the translational optical flow shows different patterns in different environments, as the information are changed because of the camera movement. This indicates that the estimation performance lowers in the case of different environments. However, our transfer learning method could solve this problem by retraining newly collected data without any label.

5. Conclusion

In this paper, we proposed a self-supervised learning approach for rotation estimation of a spherical camera. In general, fully supervised learning approaches require a large amount of labeled data, which are difficult to acquire. By contrast, our self-supervised learning approach can accomplish the training without using any labeled data. This approach is unique to spherical cameras owing to their property that optical flow can be derotated for decoupling rotational and translational optical flow components. For the regression of the camera rotation, we adopted the optical flow moment, which comprises the derotated optical flow. We experimentally confirmed that the estimation error of our approach was decreased comparing to the previous SfMLearner approach, and that the performance of our approach was comparable with that of the fully supervised learning approach. This implies that our approach could effectively estimate the camera rotation without using any labeled data. In addition, several ablation studies demonstrated that the batch normalization contributed to the improvement of the estimation performance and that the optical flow acted as robust training data rather than raw images. Finally, transfer learning with newly captured datasets was conducted to confirm the performance improvement.

In this paper, the optical flow was calculated using two equirectangular images, which were projected on a plane. However, this optical flow could be directly calculated on a sphere because spherical images are captured using spherical cameras. Also, an optical flow calculation network combined with our rotation estimation network should be designed to directly optimize our network. These will be our future works for improving the estimation performance.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

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

Notes on contributors

Dabae Kim received his B.E. degree in the Department of Mechanical Engineering and Materials Science, Yokohama National University, Japan, in 2018. Later he received his M.E. degree in the Department of Precision Engineering, the University of Tokyo, Japan, in 2020. He is currently a research engineer in ROBOTIS, Republic of Korea. His main interests are deep learning, visual odometry, and 360° image processing using spherical cameras.

Sarthak Pathak received his B.T., and M.T. degrees in the Department of Engineering Design, Indian Institute of Technology Madras, India, in 2014. He received his Ph.D. in the Department of Precision Engineering, the University of Tokyo, Japan, in 2017. He is currently a Project Assistant Professor in the Department of Precision Engineering, the University of Tokyo. His main research interest is in topics involving robot vision, specifically, localization, 3D reconstruction, and SLAM, especially using 360° spherical cameras.

Alessandro Moro received his B.S. degree in the Department of Computer Science, the University of Udine, Italy, in 2006. He received his Ph.D. degree in the Department of Computer Software Engineering, the University of Trieste, Italy, in 2011. He is a Visiting Research Fellow on Computer Vision of Chuo University, Japan, since 2011. Since 2012, he has been a Visiting Research Fellow on Computer Vision of The University of Tokyo. He is CTO at Ritecs Inc. since 2013. His research interests span computer and human vision, computer graphics, 3D reconstruction, and machine learning. Main interests are human and object recognition and machine learning for robotic application.

Atsushi Yamashita received his 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 in the RIKEN (Institute of Physical and Chemical Research). From 2001 to 2008, he was an Assistant Professor of Shizuoka University. From 2006 to 2007, he was a Visiting Associate of California Institute of Technology. From 2008 to 2011, he was an Associate Professor at Shizuoka University. From 2011, he is an Associate Professor in the Department of Precision Engineering, the University of Tokyo. His research interests include robot vision, image processing, multiple mobile robot system, and motion planning. He is a member of ACM, IEEE, JSPE, RSJ, IEICE, JSME, IEEJ, IPSJ, ITE, and SICE.

Hajime Asama received M. S., and Dr. Eng. from UTokyo (the University of Tokyo) in 1984 and 1989. He worked at RIKEN, Japan from 1986 to 2002, became a professor of RACE (Research into Artifacts, Center for Engineering) of UTokyo in 2002, a professor of School of Engineering since 2009, and the director of RACE since 2019. He received SICE System Integration Division System Integration Award for Academic Achievement in 2010, JSME Award (Technical Achievement) in 2018, etc. He was the vice-president of RSJ in 2011-2012, an AdCom member of IEEE Robotics & Automation Society in 2007-2009. Currently, he is the president of IFAC since 2020. He is a council member of the Science Council of Japan since 2017. He is a Fellow of IEEE, JSME, and RSJ.

ORCID

Dabae Kim https://orcid.org/0000-0003-4854-0658
Sarthak Pathak https://orcid.org/0000-0002-5271-1782
Alessandro Moro https://orcid.org/0000-0001-8711-0330
Atsushi Yamashita https://orcid.org/0000-0003-1280-069X
Hajime Asama https://orcid.org/0000-0002-9482-497X

References

[1] Cheng J, Sun Y, Meng MQH. Improving monocular visual SLAM in dynamic environments: an optical-flow-based approach. Advanced Robotics. 2019;33(12):576–589.
[2] Ha H, Oh TH, Kweon IS. A closed-form solution to rotation estimation for structure from small motion. IEEE Signal Processing Letters. 2018;25(3):393–397.
[3] Clark R, Wang S, Wen H, Markham A, Trigoni N. VINet: Visual-inertial odometry as a sequence-to-sequence learning problem. In: Proceedings of the 31th aaai conference on artificial intelligence (aaai). 2017. p. 3995–4001.
[4] Wang S, Clark R, Wen H, Trigoni N. DeepVO: Towards end-to-end visual odomtery with deep recurrent convolutional neural networks. In: Proceedings of the 2017 ieee international conference on robotics and automation (icra). 2017. p. 2043–2050.
[5] Kim D, Pathak S, Moro A, Komatsu R, Yamashita A, Asama H. E-CNN: Accurate spherical camera rotation estimation via uniformization of distorted optical flow fields. In: Proceedings of the 44th ieee international conference on acoustics, speech, and signal processing (icassp). 2019. p. 2232–2236.
[6] Huang AS, Bachrach A, Henry P, Krainin M, Maturana D, Fox D, Roy N. Visual odometry and mapping for autonomous flight using an RGB-D camera. In: International symposium on robotics research (isrr). 2011.
[7] Schubert D, Goll T, Demmel N, Usenko V, Stückler J, Cremers D. The TUM VI benchmark for evaluating visual-inertial odometry. In: Proceedings of the 2018 ieee/rsj international conference on intelligent robots and systems (iros). 2018. p. 1680–1687.
[8] Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research (IJRR). 2013;32(11):1231–1237.
[9] Franke U, Pfeiffer D, Rabe C, Knoeppel C, Enzweiler M, Stein F, Herrtwich RG. Making bertha see. In: Proceedings of the 2013 ieee international conference on computer vision workshops (iccvw). 2013. p. 214–221.
[10] Caron G, Morbidi F. Spherical visual gyroscope for autonomous robots using the mixture of photometric potentials. In: Proceedings of the 2018 ieee international conference on robotics and automation (icra). 2018. p. 820–827.
[11] Ye M, Li J, Ma AJ, Zheng L, Yuen PC. Dynamic graph co-matching for unsupervised video-based person re-identification. IEEE Transactions on Image Processing. 2019;28(6):2976–2990.
[12] Song J, Zhang H, Li X, Gao L, Wang M, Hong R. Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Transactions on Image Processing. 2018;27(7):3210–3221.
[13] Wang Y, Mei J, Zhang L, Zhang B, Zhu P, Li Y, Li X. Self-supervised feature learning with crf embedding for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing. 2019;57(5):2628–2642.
[14] Lee M, Fowlkes CC. CeMNet: Self-supervised learning for accurate continuous ego-motion estimation. In: Proceedings of the 2019 ieee conference on computer vision and pattern recognition workshops (cvprw). 2019.
[15] Wang FE, Hu HN, Cheng HT, Lin JT, Yang ST, Shih ML, Chu HK, Sun M. Self-supervised learning of depth and camera motion from 360° videos. In: Proceedings of the 14th asian conference on computer vision (accv). 2018. p. 53–68.
[16] Yin Z, Shi J. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the 2018 ieee conference on computer vision and pattern recognition (cvpr). Vol. 2. 2018. p. 1983–1992.
[17] Iyer G, Murthy JK, Gupta G, Krishna KM, Paull L. Geometric consistency for self-supervised end-to-end visual odometry. In: Proceedings of the 2018 ieee conference on computer vision and pattern

recognition workshops (cvprw). 2018. p. 380–388.
[18] Yu C, Liu Z, Liu XJ, Xie F, Yang Y, Wei Q, Fei Q. DS-SLAM: A semantic visual SLAM towards dynamic environments. In: Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros). 2018. p. 1168–1174.
[19] Solin A, Cortes S, Rahtu E, Kannala J. PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation. In: Proceedings of the ieee winter conference on applications of computer vision (wacv). 2018. p. 616–625.
[20] Zhang Z, Rebecq H, Forster C, Scaramuzza D. Benefit of large field-of-view cameras for visual odometry. In: Proceedings of the 2016 ieee international conference on robotics and automation (icra). 2016. p. 801–808.
[21] Forster C, Zhang Z, Gassner M, Werlberger M, Scaramuzza D. SVO: Semidirect visual odometry for monocular and multicamera systems. IEEE Transactions on Robotics. 2017;33(2):249–265.
[22] Matsuki H, von Stumberg L, Usenko V, Stückler J, Cremers D. Omnidirectional DSO: Direct sparse odometry with fisheye cameras. IEEE Robotics and Automation Letters. 2018;3(4):3693–3700.
[23] Pathak S, Moro A, Fujii H, Yamashita A, Asama H. 3d reconstruction of structures using spherical cameras with small motion. In: Proceedings of the 2016 16th international conference on control, automation and systems (iccas). 2016. p. 117–122.
[24] Da Silveira TL, Dal'Aqua LP, Jung CR. Indoor depth estimation from single spherical images. In: Proceedings of the 2018 25th ieee international conference on image processing (icip). 2018. p. 2935–2939.
[25] Su YC, Grauman K. Learning spherical convolution for fast features from 360° imagery. In: Proceedings of the 31st conference on neural information processing systems (nips). 2017. p. 529–539.
[26] Khasanova R, Frossard P. Graph-based classification of omnidirectional images. In: Proceedings of the 2017 ieee international conference on computer vision workshop (iccvw). 2017. p. 860–869.
[27] Weinzaepfel P, Revaud J, Harchaoui Z, Schmid C. DeepFlow: Large displacement optical flow with deep matching. In: Proceedings of the 2013 ieee international conference on computer vision (iccv). 2013. p. 1385–1392.
[28] Revaud J, Weinzaepfel P, Harchaoui Z, Schmid C. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the 2015 ieee conference on computer vision and pattern recognition (cvpr). 2015. p. 1164–1172.
[29] Wulff J, Sevilla-Lara L, Black MJ. Optical flow in mostly rigid scenes. In: Proceedings of the ieee conference on computer vision and pattern recognition (cvpr). 2017. p. 4671–4680.
[30] Constante G, Mancini M, Valigi P, Ciarfuglia TA. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. IEEE Robotics and Automation Letters. 2016;1(1):18–25.
[31] Mayer N, Ilg E, Hausser P, Fischer P, Cremers D, Dosovitskiy A, Brox T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the 2016 ieee conference on computer vision and pattern recognition (cvpr). 2016.
[32] Guo F, He Y, Guan L. Deep camera pose regression using motion vectors. In: Proceedings of the 25th ieee international conference on image processing (icip). 2018. p. 4073–4077.
[33] Hermes C, Einhaus J, Hahn M, Wöhler C, Kummert F. Vehicle tracking and motion prediction in complex urban scenarios. In: 2010 ieee intelligent vehicles symposium. 2010. p. 26–33.
[34] Walker J, Gupta A, Hebert M. Dense optical flow prediction from a static image. In: Proceedings of the 2015 ieee international conference on computer vision (iccv). 2015.
[35] Pathak S, Moro A, Yamashita A, Asama H. A decoupled virtual camera using spherical optical flow. In: Proceedings of the 2016 ieee international conference on image processing (icip). 2016. p. 4488–4492.
[36] Zhou T, Brown M, Snavely N, Lowe DG. Unsupervised learning of depth and ego-motion from video. In: Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr). 2017. p. 6612–6619.
[37] Godard C, Mac Aodha O, Firman M, Brostow GJ. Digging into self-supervised monocular depth prediction. In: Proceedings of the 2019 ieee international conference on computer vision (iccv). 2019. p. 3828–3838.
[38] Kim D, Pathak S, Moro A, Yamashita A, Asama H. SelfSphNet: Motion estimation of a spherical camera via self-supervised learning. IEEE Access. 2020;8:41847–41859.
[39] Jiao J, Jiao J, Mo Y, Liu W, Deng Z. Magicvo: An end-to-end hybrid cnn and bi-lstm method for monocular visual odometry. IEEE Access. 2019;7:94118–94127.
[40] Kendall A, Grimes M, Cipolla R. Posenet: A convolutional network for real-time 6-dof camera relo-

calization. In: Proceedings of the 2015 ieee international conference on computer vision (iccv). 2015. p. 2938–2946.
[41] Mahendran S, Ali H, Vidal R. 3D pose regression using convolutional neural networks. In: Proceedings of the 2017 ieee international conference on computer vision workshops (iccvw). 2017. p. 2174–2182.
[42] Gluckman J, Nayar SK. Ego-motion and omnidirectional cameras. In: Proceedings of the sixth international conference on computer vision. 1998. p. 999–1005.
[43] Ioffe S, Szegedy C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning (icml). Vol. 37. 2015. p. 448–456.
[44] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (icml). 2010. p. 807–814.
[45] Blender.org. (cited 2020 march 10).. 2020. available from https://www.blender.org/.
[46] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the 3rd international conference on learning representations (iclr). 2015.
[47] Butler DJ, Wulff J, Stanley GB, Black MJ. A naturalistic open source movie for optical flow evaluation. In: Proceedings of the 12th european conference on computer vision (eccv). 2012. p. 611–625.
[48] Turbosquid: Royalty free license, all extended uses. (cited 2020 march 10).. 2020. available from https://blog.turbosquid.com/royalty-free-license/.