IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 December 2022, accepted 25 February 2023, date of publication 6 March 2023, date of current version 14 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252884

RESEARCH ARTICLE

Scale-Aware Visual-Inertial Depth Estimation and Odometry Using Monocular Self-Supervised Learning

CHUNGKEUN LEE[1], (Graduate Student Member, IEEE), CHANGHYEON KIM[2], (Graduate Student Member, IEEE), PYOJIN KIM[3], (Member, IEEE), HYEONBEOM LEE[4], (Member, IEEE), AND H. JIN KIM[5], (Member, IEEE)

[1]Institute of Advanced Aerospace Technology, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[2]Automation and Systems Research Institute, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[3]Department of Mechanical Systems Engineering, Sookmyung Women's University, Yongsan-gu, Seoul 04312, South Korea
[4]School of Electronic and Electrical Engineering, Kyungpook National University, Buk-gu, Daegu 37224, South Korea
[5]Department of Mechanical and Aerospace Engineering, Seoul National University, Gwanak-gu, Seoul 08826, South Korea

Corresponding author: H. Jin Kim (hjinkim@snu.ac.kr)

This work was supported by the Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, Republic of Korea, under Grant NRF-2020M3C1C1A01086411.

ABSTRACT For real-world applications with a single monocular camera, scale ambiguity is an important issue. Because self-supervised data-driven approaches that do not require additional data containing scale information cannot avoid the scale ambiguity, state-of-the-art deep-learning-based methods address this issue by learning the scale information from additional sensor measurements. In that regard, inertial measurement unit (IMU) is a popular sensor for various mobile platforms due to its lightweight and inexpensiveness. However, unlike supervised learning that can learn the scale from the ground-truth information, learning the scale from IMU is challenging in a self-supervised setting. We propose a scale-aware monocular visual-inertial depth estimation and odometry method with end-to-end training. To learn the scale from the IMU measurements with end-to-end training in the monocular self-supervised setup, we propose a new loss function named as preintegration loss function, which trains scale-aware ego-motion by comparing the ego-motion integrated from IMU measurement and predicted ego-motion. Since the gravity and the bias should be compensated to obtain the ego-motion by integrating IMU measurements, we design a network to predict the gravity and the bias in addition to the ego-motion and the depth map. The overall performance of the proposed method is compared to state-of-the-art methods in the popular outdoor driving dataset, i.e., KITTI dataset, and the author-collected indoor driving dataset. In the KITTI dataset, the proposed method shows competitive performance compared with state-of-the-art monocular depth estimation and odometry methods, i.e., root-mean-square error of 5.435 m in the KITTI Eigen split and absolute trajectory error of 22.46 m and 0.2975 degrees in the KITTI odometry 09 sequence. Different from other up-to-scale monocular methods, the proposed method can estimate the metric-scaled depth and camera poses. Additional experiments on the author-collected indoor driving dataset qualitatively confirm the accurate performance of metric-depth and metric pose estimations.

INDEX TERMS Deep learning, monocular depth estimation, self-supervised learning, visual-inertial odometry.

I. INTRODUCTION

Ego-motion estimation and 3d reconstruction with a monocular camera have broad applicability because a monocular camera is inexpensive and lightweight. Especially,

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

data-driven monocular depth estimation has received attention, because they give a dense depth map from an image and ground-truth depth in a supervised manner by training a deep neural network [1], [2], [3].

To avoid the cost of collecting the ground-truth depth with an additional device, self-supervised monocular depth estimation has been proposed. State-of-the-art self-supervised methods jointly train the depth map and ego-motion for structure from motion during the training step, so it requires the sequences of monocular images during the training step [4], [5], [6], [7].

Nevertheless, self-supervised methods have scale ambiguity originating from the nature of the monocular camera because they have no criteria about the scale information unlike supervised methods with ground-truth depth information. In general, an additional sensor is introduced to address the scale ambiguity issue. In that regard, inertial measurement unit (IMU) is a popular sensor for various mobile platforms because of its lightweight and inexpensiveness. In classical vision, state-of-the-art visual-inertial methods predict the ego-motion with scale prediction using the sequences of monocular images and IMU measurements [8], [9].

Because the deep-learning-based approach has a capability of predicting a dense depth map from a *single* image, some researchers have incorporated IMU into the deep-learning like the classical visual navigation literature [10], [11], [12], [13], [14], [15]. However, learning the scale from IMU is challenging in self-supervised setting. To overcome this issue, the training concept to learn the scale from the classical visual-inertial navigation was introduced [16], [17], [18], but it highly relies on the performance of the classical navigation.

In this paper, we propose deep-learning-based depth estimation and odometry using visual-inertial data with a monocular self-supervised setup, scale-aware prediction, and an end-to-end training framework. The proposed method conserves the characteristic of the monocular self-supervised setup, which can train the network using the same type of data used for the inference. In addition, the proposed method can predict the scale-aware depth and pose with end-to-end frameworks. In comparison to the other self-supervised monocular methods, some methods need the teaching of classical visual-inertial navigation [16], [17], [18], and others only predict the up-to-scale information even if IMU measurements are provided [13], [14]. Also, unlike the classical visual-inertial navigation methods requiring the sequences of images, the proposed method can function with only a single image or two images with the IMU measurements once the network was trained.

To train the scale-aware depth and pose, we propose a new loss function named preintegration loss function, which explicitly considers the scale of the ego-motion. The scale-aware ego-motion is trained from the ego-motion by integrating the IMU measurements using the preintegration loss. Information on velocity, gravity, and bias is necessary in order to obtain the ego-motion with IMU integration. We propose the estimation method for the velocity, and the network architecture to predict the gravity in body coordinates and the bias in addition to the ego-motion and depth map for IMU integration. Also, the regulation loss function is proposed to regulate the gravity and bias and avoid the gravity and bias affected by preintegration loss only.

In addition, we propose the augmentation method for visual-inertial data. Horizontal flip augmentation is a common technique in visual learning problems due to its simplicity. However, IMU measurements should be considered if the image is flipped, so flip augmentation has rarely been performed for the visual-inertial learning problem. We propose horizontal flip augmentation considering IMU dynamics by justifying integrated ego-motion in the horizontally flipped camera coordinates.

The contributions of this paper can be summarized as the following:

- We propose a new loss function to learn the scale from the IMU measurements during the training step, which conserves the monocular self-supervised setup with the end-to-end training framework.
- We propose a network architecture predicting the gravity in body coordinates and the bias of IMU and proper regularization function for the gravity and bias to train the network using the new loss function.
- We propose a horizontal flip augmentation method for visual-inertial data considering IMU dynamics.

We validate the proposed method in the KITTI dataset and an indoor experiment in comparison with the state-of-the-art deep-learning-based monocular self-supervised methods and classical monocular visual-inertial navigation methods.

We describe related works in section II. Then, we briefly describe self-supervised monocular depth estimation and IMU preintegration as preliminary in section III. Section IV describes the proposed method including the preintegration loss function and the network architecture. Then, we perform an ablation study and validate the proposed algorithm by comparison with other methods in section V. Finally, conclusion follows.

II. RELATED WORKS

In this section, we list related works about deep-learning-based monocular depth estimation and classical visual-inertial navigation. First, deep-learning-based depth estimation with monocular images is categorized based on the training data. Then, the classical visual-inertial navigation is discussed. Finally, we summarize the deep-learning-based depth estimation and odometry which receive an image and IMU as input.

A. SUPERVISED MONOCULAR METHODS

Supervised methods aim to construct a deep neural network that predicts the depth map from a single RGB image and the ground-truth depth information. Such possibility was firstly reported in [1] with the convolutional neural

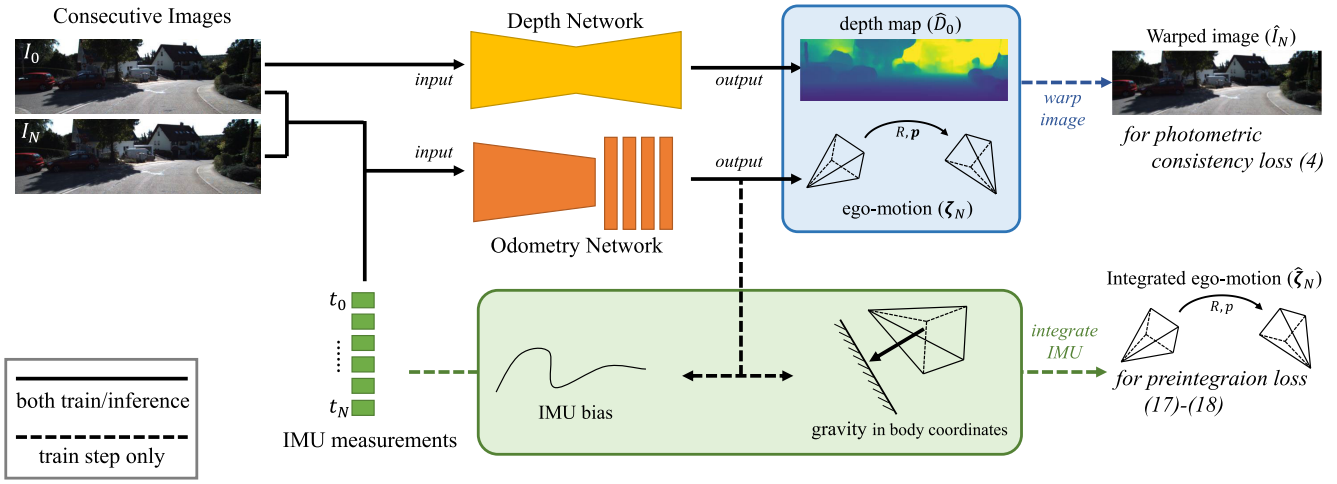


FIGURE 1. The overview of the proposed algorithm. During the inference step, a dense depth map and ego-motion are predicted from a pair of consecutive images and IMU measurements. During the training step, the gravity direction in the body coordinate and the bias of IMU are additionally predicted to learn the real-world scale for the preintegration loss function.

network. By adopting the fully convolutional neural network, the monocular depth estimation was significantly improved [2]. Many researchers have focused on adding more depth cues or refining training problems during the training step for an accurate depth map [3], [19], [20]. The supervised methods show better performance than unsupervised or self-supervised methods, but they need ground-truth depth information during the training step.

B. UNSUPERVISED MONOCULAR METHODS

Unsupervised methods aim to train the depth estimation network without the ground-truth depth or pose information. To learn the depth with no ground-truth depth data, the pairs of stereo images have been employed. The reconstruction error was proposed in [21], which originated from the epipolar geometry. This loss was refined in [22] to a fully differentiable form for better backpropagation. Some research focused on adding depth cues like supervised approaches, such as the consistency of the disparity between the left and right images proposed in [22]. Meanwhile, the work in [23] formulated the unsupervised stereo approach as the synthesizing and stereo-matching problem. Those unsupervised methods can predict the depth map with the scale information, but they require a stereo system for collecting training data.

C. SELF-SUPERVISED MONOCULAR METHODS

Self-supervised methods aim to train the depth estimation network with sequences of monocular or stereo images. In this paper, we focus on self-supervised monocular methods, which utilize the sequences of monocular images. It is noted that self-supervised monocular methods require monocular images during the training step. Thus, the network can be trained from the data collected during the inference. State-of-the-art self-supervised monocular methods jointly estimate the depth map and ego-motion [4]. Some approaches have tried to predict optical flow in addition to the depth information to explicitly express geometry information [5], [24]. Reference [25] designed the photometric consistency loss in the latent space obtained from the auto-encoder network instead of raw images. Considering the 3-dimensional property, [26] proposed the 3D packing network, which performs 3D convolution operation from the 2D data. Edge-aware depth prediction with high-resolution images was performed [27]. Focusing on scale consistency, [28] proposed scale-aware geometric loss by aligning the point clouds between frames, which helped to conserve the scale but could not predict the metric scale.

Some research focused on real-time applications for mobile platforms by optimizing the network architecture [29], [30]. To reject occluded or moving pixels for the reprojection loss, the auto-mask was introduced in [7], which generates a binary mask of the possibly occluded region. Reference [31] adopted the generative adversarial networks into the monocular depth estimation, and showed the performance enhancement. For the input of the sequence of images, the recurrent neural network was proposed in [6], [32]. However, these self-supervised methods suffer from scale ambiguity due to the nature of the monocular camera.

D. CLASSICAL VISUAL-INERTIAL METHODS

Classical visual-inertial navigation, for odometry and simultaneous localization and mapping (SLAM), has been widely researched. Some survey papers provide a good review of its long history [33], [34], [35]. Thus, in this paper, we only mention the common characteristics and a few state-of-the-art monocular visual-inertial navigation methods.

Classical navigation methods can be categorized into two types based on the problem formulation: filtering-based and optimization-based methods. Filtering-based methods design the filter which expresses the state and measurements of

the robot, respectively. Then, the designed filter is operated during inference [36], [37], [38], [39]. Optimization-based methods construct the performance index from the camera geometry and perform non-linear optimization [8], [9], [40].

For the optimization-based methods, IMU initialization is required at the beginning, and the vehicle/robot should generate the acceleration and the tilting motion in the roll and pitch directions for the monocular case. In automobile environments, IMU initialization may fail because the dominant motion of the car is yaw direction.

E. LEARNING-BASED VISUAL-INERTIAL METHODS

Some researchers included IMU into the deep-learning approach when the ground-truth information is given. Reference [10] showed that visual-inertial odometry can be solved using a deep-learning-based approach. For the robustness against noisy image or IMU observation, the selective fusion layer which fuses the visual and inertial features was proposed in [11]. However, both methods require the ground-truth pose during the training step.

Unsupervised visual-inertial odometry with the pipeline of IMU preintegration was proposed in [12], but it requires stereo images during the training step. Reference [13] proposed self-supervised monocular visual-inertial depth estimation and odometry, adopting the generative adversarial network. The predicted poses are integrated and additionally optimized using the geometric and trajectory consistency with monocular-inertial input [14]. Both methods, however, cannot predict the scale information. A multi-level visual-inertial odometry strategy was proposed in [15] for RGB-D images. They can predict scale, but they need the dense depth map as an input containing the scale information.

To estimate the scale with IMU measurements, some methods tried operating classical visual-inertial odometry to obtain the sparse depth or ego-motion having scale information. The depth completion using the sparse depth provided by the classical visual-inertial navigation was proposed in [16], [17]. Those works, nonetheless, assume that sparse depth information is given by a navigation system for depth prediction, which requires a sequence of images, not a single image. Reference [18] proposed the transfer learning of the depth estimation network with teaching of the classical visual-inertial method to learn the scale of the new environment. Since these methods require classical navigation during the training step, whenever the classical method fails, so do they.

In this paper, we propose the scale-aware monocular self-supervised method using IMU measurements with end-to-end training. All the other algorithms discussed above did not achieve at least one of those characteristics: the monocular self-supervised setup, scale-aware prediction, and end-to-end training. Some researchers provided scale-aware prediction with the supervised, unsupervised, or stereo self-supervised setup requiring additional training data collected from LiDAR, stereo system, and so on [10], [11], [12], [15]. Others predicted up-to-scale depth with monocular self-supervised setup with the IMU measurements [13], [14]. The others could predict scale-aware depth by teaching from the classical visual-inertial navigation method, which could not train in an end-to-end manner and relied on the performance of the classical visual-inertial navigation method [16], [17], [18].

III. PRELIMINARY

The proposed method originates from self-supervised monocular depth estimation and is upgraded to learn the scale by integrating IMU measurements. In this section, we describe the self-supervised monocular depth estimation and IMU preintegration as preliminaries.

A. SELF-SUPERVISED MONOCULAR DEPTH ESTIMATION

For self-supervised monocular depth estimation, the depth map and the ego-motion are jointly predicted. Thus, two convolutional neural networks are constructed: one is the depth network predicting the dense depth map from a single RGB image, and the other is the pose network predicting the relative pose from a pair of consecutive images.

To train the networks, the photometric consistency loss is introduced based on the motion stereo. To express the motion stereo, from the point in the pixel coordinate of the current view $\boldsymbol{u}_n$, we obtain the point in the pixel coordinate of the next view $\hat{\boldsymbol{u}}_{n+1}$ using a predicted dense depth map $\hat{D}_n$, predicted relative pose $\hat{T}_{n\to n+1}$ in the special Euclidean group SE(3) and camera intrinsic parameter K as

$$\hat{\boldsymbol{u}}_{n+1} = K\hat{T}_{n\to n+1}\hat{D}_n(\boldsymbol{u}_n)K^{-1}\boldsymbol{u}_n \tag{1}$$

with some notation abuses for simplicity.

Then, we warp the current image I_n into the view of the next frame denoted as $\hat{I}_{n+1}$ by the differentiable bilinear sampling mechanism [41]. The photometric consistency loss [42] is formulated as the distance between the target image I_{n+1} from the dataset and the warped image $\hat{I}_{n+1}$ as

$$\mathcal{L}_{\text{photo}} = \frac{1}{N}\sum_{\boldsymbol{u}}\left[\text{dist}(I_{n+1}, \hat{I}_{n+1})\right] \tag{2}$$

where $\text{dist}(x, y)$ is a distance function between two images. In this paper, we adopt [42] given as $\text{dist}(x, y) = \alpha\,|x - y| + 0.5(1-\alpha)(1-\text{SSIM}(x, y))$ with the scalar constant $\alpha = 0.15$.

In addition, for regulation, we also minimize edge-aware depth smoothness [22] as

$$\mathcal{L}_{\text{smooth}} = \sum_{i\in\{x,y\}} |\partial_i\hat{D}_n| \exp\left(-|\partial_i I_n|\right) \tag{3}$$

where ∂_i is the partial derivative operator respective to i.

To reject the effect of the occluded or moving pixel which breaks the photometric consistency, per-pixel mask $\mu \in [0, 1]$ is multiplied by the distance between images for the photometric consistency loss as

$$\mathcal{L}_{\text{photo}} = \frac{1}{N}\sum_{\boldsymbol{u}}\left[\mu\ \text{dist}(I_{n+1}, \hat{I}_{n+1})\right]. \tag{4}$$

In this paper, we adopt the auto-mask [7], binary masking method calculated from source and target images as

$$\mu = \begin{cases} 1 & \text{if dist}(I_{n+1}, \hat{I}_{n+1}) > \text{dist}(I_{n+1}, I_n) \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

B. IMU PREINTEGRATION

IMU preintegration aims to integrate raw IMU measurements to obtain the ego-motion between two image frames. In other words, the rotation R_N, velocity v_N and translation p_N at the next frame ego-motion ($\bullet_N$) should be formulated with the current ego-motion ($\bullet_0$) and IMU measurements containing the acceleration $\tilde{a}_i$ and the angular velocity $\tilde{\omega}_i$.

Then, the ego-motion at the i-th frame on the inertial coordinate is expressed as

$$R_i = R_0 \prod_{k=0}^{i-1} \exp\left(\boldsymbol{\omega}_k \Delta t_k\right) \tag{6}$$

$$\boldsymbol{v}_i^G = \boldsymbol{v}_0^G + \sum_{k=0}^{i-1} (R_k \boldsymbol{a}_k - \boldsymbol{g}^G) \Delta t_k \tag{7}$$

$$\boldsymbol{p}_i^G = \boldsymbol{p}_0^G + \sum_{k=0}^{i-1} \boldsymbol{v}_k \Delta t_k + \frac{1}{2} \sum_{k=0}^{i-1} (R_k \boldsymbol{a}_k - \boldsymbol{g}^G) \Delta t_k^2 \tag{8}$$

where $\bullet^G$ is the variable in the inertial coordinate, Δt_k is the elapsed time from k to $k+1$, $\boldsymbol{\omega}_k = \tilde{\boldsymbol{\omega}}_k - \boldsymbol{b}_k^{\boldsymbol{\omega}} - \boldsymbol{\eta}_k^{\boldsymbol{\omega}}$ is the unbiased angular velocity at k with bias $\boldsymbol{b}_k^{\boldsymbol{\omega}}$ and noise $\boldsymbol{\eta}_k^{\boldsymbol{\omega}}$, $\boldsymbol{a}_k = \tilde{\boldsymbol{a}}_k - \boldsymbol{b}_k^a - \boldsymbol{\eta}_k^{\boldsymbol{a}}$ is the unbiased acceleration at k with bias $\boldsymbol{b}_k^{\boldsymbol{a}}$, and the gravity $\boldsymbol{g}^G$. exp is an exponential map of the Lie algebra of the special orthogonal group so(3).

IV. SELF-SUPERVISED MONOCULAR VISUAL-INERTIAL DEPTH ESTIMATION AND ODOMETRY

In this section, we describe the proposed method which learns the real-world scale from the IMU measurements. Fig. 1 shows the overview and flowchart of the proposed method. The proposed method predicts the depth map from the single image, and predicts the ego-motion, the gravity in body coordinates, and IMU bias from the two consecutive images and IMU measurements. Next, the image is warped using the depth map and ego-motion to obtain and minimize the photometric consistency loss defined as the difference between the reference image and the warped image. In addition, the IMU measurements are integrated using the predicted gravity and bias to obtain the scale-aware ego-motion. Then, we minimize the preintegration loss defined as the difference between the integrated ego-motion and prediction ego-motion. The integrated ego-motion has scale, so does the predicted ego-motion by minimizing the preintegration loss.

We formulate the proposed method into two parts. The first part is the loss function to train the network, which generates the relation about the scale from the IMU measurements during the training step. The second part is the network architecture suitable for the proposed loss function.

A. LOSS FUNCTION

We describe each loss function of the proposed method, and the total loss is described at the end of this section. The proposed methods utilize three loss functions to optimize networks: photometric consistency loss function from state-of-the-art self-supervised monocular depth estimation to learn scale-unaware depth and ego-motion, preintegration loss function to learn the scale of ego-motion from the IMU measurements, and regulation loss function about the gravity direction and the bias to regulate the effect of predicted gravity direction and the bias.

1) PHOTOMETRIC CONSISTENCY LOSS

The photometric consistency loss has been widely employed to optimize the depth map and the ego-motion from consecutive images. This loss expresses the epipolar geometry structure from motion. We adopt the photometric consistency loss (4) and the depth smoothness loss (3) like most state-of-the-art self-supervised methods described in section III-A.

2) PREINTEGRATION LOSS

The preintegration loss obtains the scale-aware ego-motion by integrating IMU measurements like IMU preintegration described in section III-B and compares predicted ego-motion with the obtained ego-motion. The main role of the preintegration loss is to learn the scale from IMU measurements by integrating the IMU measurements and correcting integrated and predicted ego-motion. Since the photometric consistency loss does not incorporate the scale, only preintegration loss contributes to the learning of the scale.

For the predicted relative pose in the body coordinate $\boldsymbol{\zeta}_N = (\boldsymbol{w}_N, \boldsymbol{z}_N)$ defined on se(3), the Lie algebra of SE(3), at time N, we define the relative form of the rotation ΔR_N, the velocity $\Delta \boldsymbol{v}_N$ and the translation $\Delta \boldsymbol{p}_N$ as

$$\Delta R_N := R_0^T R_N = \exp \boldsymbol{w}_N \tag{9}$$

$$\Delta \boldsymbol{v}_N := R_0^T \left(\boldsymbol{v}_N^G - \boldsymbol{v}_0^G\right) = \Delta R_N \, \boldsymbol{v}_N^B - \boldsymbol{v}_0^B \tag{10}$$

$$\begin{aligned} \Delta \boldsymbol{p}_N &:= R_0^T \left(\boldsymbol{p}_N^G - \boldsymbol{p}_0^G - \boldsymbol{v}_0^G \Delta T_N\right) \\ &= \boldsymbol{p}_N^B - \boldsymbol{v}_0^B \Delta T_N \end{aligned} \tag{11}$$

where $\bullet^G$ and $\bullet^B$ are the parameter in the inertial and body coordinates, respectively, ΔT_N is the elapsed time from time 0 to time N, and $\boldsymbol{p}_N^B = J_{\boldsymbol{w}_N} \boldsymbol{z}_N$ is the translation of the relative pose with the left Jacobian of $\boldsymbol{w}_N$ denoted as $J_{\boldsymbol{w}_N}$.

Using IMU preintegration (6)-(8), $\Delta \bullet_N$ can be written as

$$\Delta \hat{R}_N = \prod_{k=0}^{N-1} \exp\left(\hat{\boldsymbol{\omega}}_k \Delta t_k\right) \tag{12}$$

$$\Delta \hat{\boldsymbol{v}}_N = \sum_{k=0}^{N-1} (\Delta \hat{R}_k \hat{\boldsymbol{a}}_k - \boldsymbol{g}_N^B) \Delta t_k \tag{13}$$

$$\Delta \hat{\boldsymbol{p}}_N = \sum_{k=0}^{N-1} \Delta \hat{\boldsymbol{v}}_k \Delta t_k + \frac{1}{2} \sum_{k=0}^{N-1} (\Delta \hat{R}_k \hat{\boldsymbol{a}}_k - \boldsymbol{g}_N^B) \Delta t_k^2 \tag{14}$$

where $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\eta}^{\omega}$ is the measured angular velocity, $\hat{\boldsymbol{a}} = \boldsymbol{a} - \boldsymbol{\eta}^{\boldsymbol{a}}$ is the measured acceleration, and $\boldsymbol{g}_N^B = R_N \boldsymbol{g}^G$ is the gravity in the body coordinate. In this paper, we denote $\Delta\bullet_N$ as $\bullet_N$ from the relative pose and $\Delta\hat{\bullet}_N$ as that from the IMU measurements to distinguish those.

From (9)-(14), three equality constraints can be generated: $\Delta\bullet_N = \Delta\hat{\bullet}_N$ where $\bullet = R, \boldsymbol{v}, \boldsymbol{p}$. To solve those equality constraints, however, the velocity $\boldsymbol{v}_0^B$ should be known. From the equality of the translation $\boldsymbol{p}$, the closed-form of the velocity can be obtained as

$$\boldsymbol{v}_0^B = \frac{1}{\Delta T_N}\left(\boldsymbol{p}_N^B - \Delta\hat{\boldsymbol{p}}_N\right). \tag{15}$$

Then, $\Delta\boldsymbol{v}_N$ is reformulated as

$$\Delta\boldsymbol{v}_N = \Delta R_N \boldsymbol{v}_N^B - \frac{1}{\Delta T_N}\left(\boldsymbol{p}_N^B - \Delta\hat{\boldsymbol{p}}_N\right). \tag{16}$$

The calculated velocity $\boldsymbol{v}_0^B$ may be noisy because (15) is the finite difference of the predicted pose $\boldsymbol{p}_N^B$. Therefore, we smooth the translation $\boldsymbol{p}_N^B$ by moving the average filter along the temporal axis when calculating the velocity to suppress the noise in the implementation.

Finally, we obtain the preintegration loss as the norm of two remaining equality constraints as

$$\mathcal{L}_{\text{rot}} = \lambda_{\text{rot}} \left\| \Delta R_N - \Delta\hat{R}_N \right\| \tag{17}$$
$$\mathcal{L}_{\text{vel}} = \lambda_{\text{vel}} \left\| \Delta\boldsymbol{v}_N - \Delta\hat{\boldsymbol{v}}_N \right\| \tag{18}$$

where $\lambda_\bullet$ is the hyper-parameter for weighting the loss functions and $\|\bullet\|$ is a norm function. We adopt the logcosh as a norm function in the implementation to suppress the effect of the outlier for fast convergence.

3) REGULATION LOSS

In addition to the scale and ego-motion, both predicted gravity and bias affect the IMU preintegration loss. If no regulation is performed, gravity and bias can be freely regressed, so the scale, gravity and bias may be wrongly estimated. Therefore, we carefully design the regulation loss of the gravity direction and the bias of IMU.

a: GRAVITY REGULATION

The gravity in the inertial coordinate is assumed constant, and the gravity in the body coordinate and the ego-motion are coupled. Hence, we design the gravity regulation loss to express the gravity in the body coordinate using the predicted ego-motion. In the body coordinate, the gravity at time N ($\hat{\boldsymbol{g}}_N^B$) can be estimated from the gravity predicted by the network at time 0 ($\boldsymbol{g}_0^B$) and the rotational ego-motion (ΔR_N) as

$$\hat{\boldsymbol{g}}_N^B = \Delta R_N \boldsymbol{g}_0^B. \tag{19}$$

Since the magnitude of gravity is constant, we adopt the geodesic distance on the surface of the sphere as the loss function between the gravity predicted by the network at time N ($\boldsymbol{g}_N^B$) and the estimated gravity $\hat{\boldsymbol{g}}_N^B$ from (19):

$$\mathcal{L}_{\text{grav}} = \lambda_{\text{grav}} \arctan \frac{\left|\boldsymbol{g}_N^B \times \hat{\boldsymbol{g}}_N^B\right|}{\boldsymbol{g}_N^B \cdot \hat{\boldsymbol{g}}_N^B} \tag{20}$$

where the symbols $\cdot$ and $\times$ are the inner and outer products defined in $\mathbb{R}^3$.

b: BIAS REGULATION

It is known that the bias varies slowly, so most classical visual-inertial navigation methods construct the bias model as a constant with Gaussian noise. Similarly, we regulate both angular and linear bias by minimizing the bias difference among adjacent frames as

$$\mathcal{L}_{\text{bdiff}} = \lambda_{\text{bdiff}\boldsymbol{\omega}} \left\| \boldsymbol{b}_N^{\boldsymbol{\omega}} - \boldsymbol{b}_0^{\boldsymbol{\omega}} \right\|_2^2 + \lambda_{\text{bdiff}\boldsymbol{a}} \left\| \boldsymbol{b}_N^{\boldsymbol{a}} - \boldsymbol{b}_0^{\boldsymbol{a}} \right\|_2^2. \tag{21}$$

In addition to the regulation among adjacent frames, we also regulate the magnitude of the bias term to avoid bias prediction that is too large. This regulation is expressed as

$$\mathcal{L}_{\text{bmag}} = \lambda_{\text{bmag}\boldsymbol{\omega}} \left\| \boldsymbol{b}_N^{\boldsymbol{\omega}} \right\|_2^2 + \lambda_{\text{bmag}\boldsymbol{a}} \left\| \boldsymbol{b}_N^{\boldsymbol{a}} \right\|_2^2. \tag{22}$$

4) TOTAL LOSS

The total loss function is the linear combination of the above losses: the photometric consistency loss (4) with the smoothness loss (3), the preintegration loss of the rotational part (17) and the velocity part (18), the gravity regulation loss (20), and the bias regulation loss (21)-(22), as

$$\begin{aligned}\mathcal{L} = \; & \mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{vel}} + \\ & \mathcal{L}_{\text{grav}} + \mathcal{L}_{\text{bdiff}} + \mathcal{L}_{\text{bmag}}.\end{aligned} \tag{23}$$

B. NETWORK ARCHITECTURE

The proposed approach estimates the depth map, the ego-motion, the gravity direction and IMU bias from images and IMU measurements. We design two networks: a depth network and an odometry network. The depth network estimates the depth map from a single RGB image. The odometry network estimates the relative pose between two frames, gravity direction and IMU bias from the two consecutive images and IMU measurements between images.

1) DEPTH NETWORK

We adopt the depth network proposed in [7]. The network has the U-Net structure [43], which is a fully convolutional encoder-decoder structure with skip-connection. We select ResNet18 [44] as an encoder. The depth network receives a single image, and no IMU information is received. The proposed depth network can estimate the scale by learning the scale using the preintegration loss during the training step.

2) ODOMETRY NETWORK

We design an odometry network emitting relative pose, IMU bias and gravity direction. Fig. 2 shows the outline of the proposed odometry network. The odometry network consists of

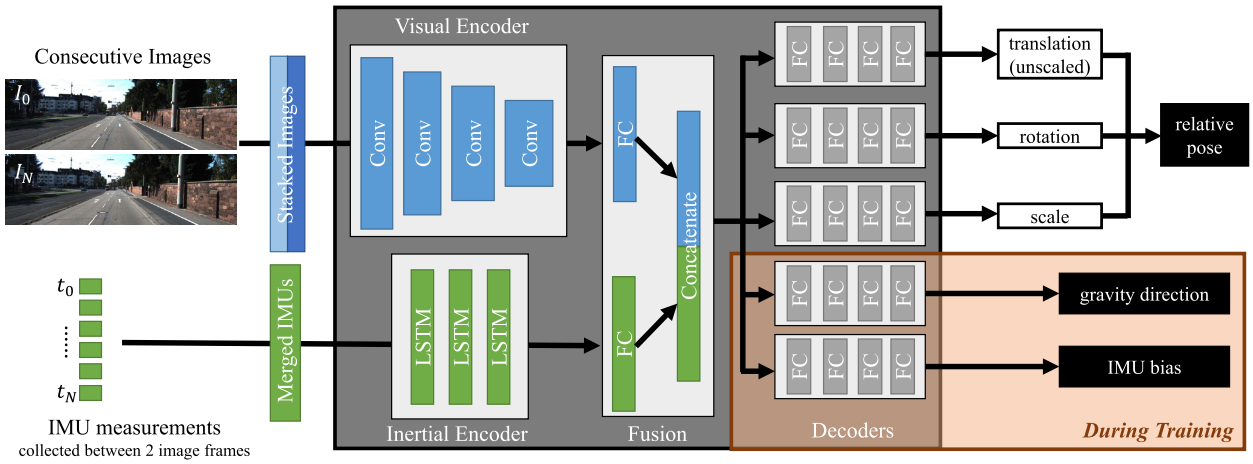


FIGURE 2. The outline of the proposed odometry network. The network receives a pair of consecutive images and IMU measurements between images. Then, the network emits the relative pose between images, the direction of gravity in the body coordinate, and the bias of the IMU measurements. Conv is the convolution layer, LSTM is the long short-term memory layer, and FC is the fully connected layer, respectively.

a visual encoder, inertial encoder, feature fusion and several decoders.

a: VISUAL ENCODER

We select ResNet18 as a visual encoder, which is almost the same as that of the depth network. As input, two consecutive images stacked along the channel axis are provided.

b: INERTIAL ENCODER

We adopt bidirectional Long Short-Term Memory (LSTM) for encoding the IMU measurements. Because IMU measurements are stacked between two image frames, they have temporal meaning so a recurrent neural network is selected. In detail, the inertial encoder has three bidirectional LSTM layers with 128 channels and one dense layer with 128 dimensions.

c: FEATURE FUSION

The feature fusion part of the network aims to merge visual and inertial features provided by each encoder. We focus on balancing each feature to avoid a feature being ignored. Firstly, we add a single dense layer for each network: 256 for a visual encoder, and 128 for an inertial encoder. Then, we normalize by layer normalization [45] for each feature to have a similar magnitude. Next, both features are concatenated to be fused as a single feature.

d: DECODER

Several decoders receive the same feature from the feature fusion part and emit the final output, respectively. Each decoder contains seven dense layers to decode the fused feature. The channel of each layer is 256, 256, 128, 128, 64, 64, and the dimension of the output (e.g., 7 for the relative pose, 2 for the gravity, and 6 for the bias). Each decoder has an additional activation to properly constrain the output. In general, the input and output of the network have a magnitude of around 1, so the scalar hyper-parameter may be multiplied into the output. Additionally, special activation is applied to some outputs if necessary.

For the gravity, we regress a 2-dof vector on the spherical coordinate because the magnitude of the gravity is fixed. For fast convergence of gravity prediction, the gravity direction is converted considering the nominal gravity direction in the robot platform by initializing the network bias as zero. For instance, if the nominal gravity direction heads z-axis like the KITTI dataset, the activation for the gravity direction is

$$\boldsymbol{g}^B = \|\boldsymbol{g}\|_2 \left[\sin\theta\cos\phi \;\; \sin\theta\sin\phi \;\; \cos\theta\right]^T \tag{24}$$

where (θ, ϕ) are inclination and azimuth, which are predicted from the network, and $\|\boldsymbol{g}\|_2 = 9.81\ \mathrm{m/s^2}$.

For the relative pose, we regress a 7-dof vector of logarithm forms of the translation $\tilde{z}$, rotation $\boldsymbol{\omega}$ and the pseudo-scale s for the ego-motion $(z, \boldsymbol{\omega})$ on se(3) as

$$z = \tilde{z}\exp s. \tag{25}$$

Here, the pseudo-scale $\exp s$ is not a real-world scale because the magnitude of the translation $\tilde{z}$ is not constrained.

V. EXPERIMENTAL VALIDATION

We perform validation in the KITTI dataset [46], one of the famous dataset collected using the vehicle, and in indoor environments with the automobile platforms at the underground parking lot. First, we perform two ablation studies to show whether the proposed loss function contributes to learning the scale. Then, we show the depth performance based on the Eigen split and the pose performance based on the KITTI odometry dataset. We show the result of the proposed method, with the description of the detailed implementation of the experiment.

A. DATA AUGMENTATION

For deep-learning applications, data augmentation has been widely performed to generate additional data from a given dataset. In this section, we describe our data augmentation.

1) IMAGE AUGMENTATION

We perform image augmentation at 50% probability, by changing the brightness, contrast, saturation and hue. If image augmentation is performed, we randomly select values from uniform distribution: brightness $\in$ [0.8, 1.2], contrast $\in$ [0.8, 1.2], saturation $\in$ [0.8, 1.2] and hue in degree $\in$ [−36, 36]. All the images in the sequence are converted with the same type of augmentation.

2) LEFT-RIGHT FLIP AUGMENTATION

Flip augmentation is a common augmentation method for deep-learning-based visual applications. Because the gravity usually heads downwards in the camera view, only a left-right direction flip is performed to conserve the nominal direction of gravity. Unlike image augmentation, IMU measurements should also be converted for this flip, since the ego-motion in the coordinate of the flipped camera is changed. We generate the corresponding IMU measurements to justify the ego-motion obtained from the integration of IMU measurements in the coordinates of the flipped camera as

$$\tilde{\boldsymbol{w}} = -T^T V T \boldsymbol{w} \tag{26}$$

$$\tilde{\boldsymbol{a}} = T^T V T \boldsymbol{a} \tag{27}$$

where $(\boldsymbol{w}, \boldsymbol{a})$ is IMU measurement with angular velocity $\boldsymbol{w}$ and acceleration $\boldsymbol{a}$, and $(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{a}})$ is the converted IMU measurement for left-right flip augmentation, $V = \mathrm{diag}(-1, 1, 1)$ is transformation of flipped dynamics, and T is the rotation part of the extrinsic calibration matrix.

B. IMPLEMENTATION DETAIL

In this section, we describe the implementation details of the networks, the loss functions, and the optimization setup.

1) NETWORK

Each output of the decoder is multiplied by the following heuristic value. noitemsep

- Angular part of the ego-motion: 1e-2
- Translation part of the ego-motion: 1e-2
- Pseudo-scale of the ego-motion: 1e0
- Gravity direction as angles: 3e-1
- IMU bias (angular part): 1e-2
- IMU bias (acceleration part): 1e-1

The proposed odometry network has three separate decoders emitting the angular part of the ego-motion, the translation part of the ego-motion and the pseudo-scale of the ego-motion. Due to this separated decoder strategy, tuning the heuristic values is almost not necessary.

2) LOSS DETAIL

To train the network, we use the ADAM [47] optimizer. The learning rate is 4e-5 at the beginning of the training and decreased by the inverse time policy with 0.98 ratios. Additionally, each loss function has the weighting parameter determined by the heuristic way as noitemsep

- Photometric consistency loss: 1
- Depth smoothness loss: 1e-2
- Preintegration loss (angular part): 4e3
- Preintegration loss (velocity part): 4e1
- Gravity regulation loss: 4e0
- Bias difference regulation loss (angular part): 1e2
- Bias difference regulation loss (acceleration part): 1e2
- Bias magnitude regulation loss (angular part): 1e-2
- Bias magnitude regulation loss (acceleration part): 1e-2

3) DATASET DETAIL

We collect all train data and randomly select approximately 1,100 sequences for each epoch. Each sequence consists of 8 consecutive images and a fixed number of IMU measurements observed between two consecutive images (e.g., 10 for the KITTI dataset, and 25 for the experiment). We iterate 200 epochs to train the networks, which takes approximately 40 hours in TITAN Xp GPU environments.

C. PERFORMANCE INDICES

In this section, we describe the performance indices for depth and odometry validation in the xy-plane considering automobile applications. For monocular methods that cannot predict the scale, the scale is directly taken from the *ground-truth*.

1) DEPTH VALIDATION

For the depth validation, five performance indices have been widely reported: absolute relative error (Abs Rel), square relative error (Sq Rel), root mean square error (RMSE), log scale root mean square error (RMSE log), and accuracy (δ). The ratio of pixels is reported whose accuracy is less than 1.25, 1.25^2, and 1.25^3, respectively.

$$\text{Abs Rel} = \mathbb{E}\left(\left|D - \hat{D}\right| / D\right) \tag{28}$$

$$\text{Sq Rel} = \mathbb{E}\left(\left(D - \hat{D}\right)^2 / D\right) \tag{29}$$

$$\text{RMSE} = \sqrt{\mathbb{E}\left(\left(D - \hat{D}\right)^2\right)} \tag{30}$$

$$\text{RMSE log} = \sqrt{\mathbb{E}\left(\left(\log D - \log \hat{D}\right)^2\right)} \tag{31}$$

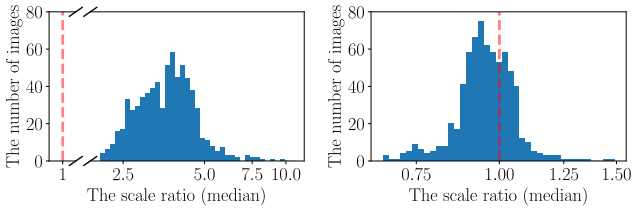
$$\text{accuracy}(\delta) = \max\left(D/\hat{D}, \hat{D}/D\right) \tag{32}$$

where D and $\hat{D}$ are the ground-truth and predicted depth, respectively, and $\mathbb{E}(\bullet)$ is the average of $\bullet$.

For the methods with no scale prediction, the scale is taken from the ground-truth depth in the same manner as [4], which is the ratio of estimated median depth to ground-truth

TABLE 1. The depth performance for the ablation study about preintegration loss function.

Methods	Scale[a]	Abs Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$
No preint	RAW	2.5265	35.208	0.0129	0.0349
	from GT	0.1560	6.9432	0.7758	0.9157
proposed	RAW	0.1408	5.4352	0.8038	0.9421
	from GT	0.1252	5.1737	0.8579	0.9530



(a) No preintegration loss (b) proposed

FIGURE 3. The histogram of the scale from the predicted depth among frames calculated by (33) for the ablation study about the preintegration loss function. If the method estimates the real-world scale, the value should be one, represented as the red-dashed line.

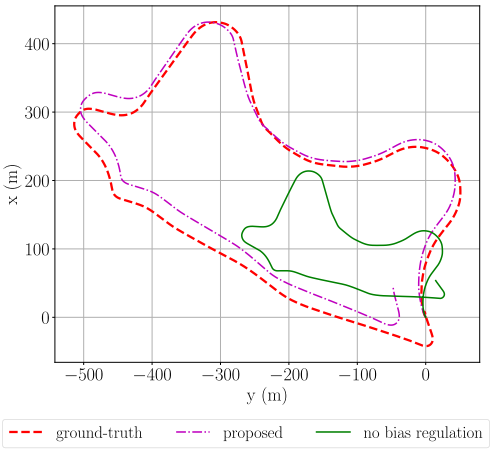


FIGURE 4. Top-down view of the predicted trajectory of KITTI odometry 09 for the ablation study concerning the bias regulation loss function.

median depth:

$$s_{\text{depth}} = \frac{\text{median}(D)}{\text{median}(\hat{D})}. \tag{33}$$

2) POSE VALIDATION

In the odometry validation, average trajectory error (ATE) and relative pose error (RPE) are some of the famous performance indices as

$$\text{ATE}_i = P_i^{-1} S \hat{P}_i \tag{34}$$

$$\text{RPE}_i = \left(P_i^{-1} P_{i+1}\right)^{-1} \left(\hat{P}_i^{-1} \hat{P}_{i+1}\right) \tag{35}$$

where P_i and $\hat{P}_i$ are the ground-truth and the predicted pose in the inertial coordinate, respectively, and S is a time-invariant rigid body transformation for the alignment. We compute

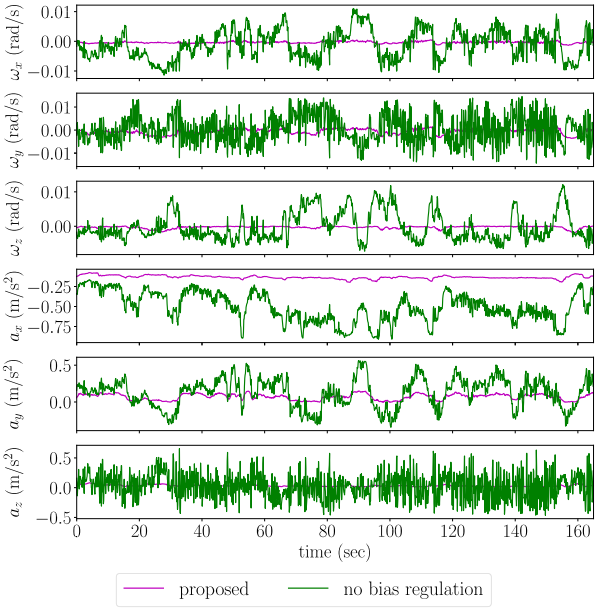
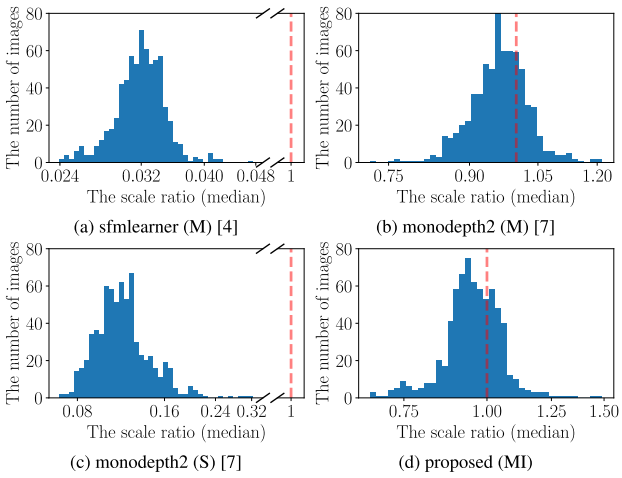


FIGURE 5. The bias prediction result at the KITTI odometry 09 for the ablation study about the regulation loss function.



(a) sfmlearner (M) [4] (b) monodepth2 (M) [7]

(c) monodepth2 (S) [7] (d) proposed (MI)

FIGURE 6. The histogram of the scale from the predicted depth among frames calculated by (33). If the method estimates the real-world scale, so the value should be one, represented as the red-dashed line. (·) next to denotes the training data: (M) is monocular sequence, (S) is stereo sequence and (MI) is monocular sequence with IMU measurement.

RMSE of the translation and the average of the rotation part as

$$*_{tr} = \text{argmin}_S \sqrt{\mathbb{E}\left(\|\bullet\|_{\text{tr}}^2\right)} \tag{36}$$

$$*_{rot} = \mathbb{E}\left(\|\bullet\|_{\text{rot}}\right) \tag{37}$$

where $*$ is ATE or RPE, $\|\bullet\|_{\text{tr}}$ is the Euclidean 2-norm of the translation part of the rigid body matrix and $\|\bullet\|_{\text{rot}}$ is the Euclidean 2-norm of the logarithm of the rotation matrix of the rigid body matrix.

For the methods with no scale prediction, a single scale value is taken from the ground-truth ego-motion across the

TABLE 2. Depth estimation performance in KITTI Eigen split (80 m cap).

Methods	Scale[a]	Train[b]	Abs Rel	Sq Rel	RMSE	RMSE^{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
monodepth [22]	Predicted	Unsup(S)	0.148	1.344	5.927	0.247	0.803	0.922	0.964
MonoGAN [31]	Predicted	Unsup(S)	0.119	1.239	5.998	0.212	0.846	0.940	0.976
SVS [23]	Predicted	Unsup(S)	0.094	0.626	4.252	0.177	0.891	0.965	0.984
monodepth2 [7]	Predicted	Selfsup(S)	0.106	0.818	4.750	0.196	0.874	0.957	0.979
Featdepth [25]	Predicted	Selfsup(S)	0.099	0.697	4.427	0.184	0.889	0.963	0.982
HR-Depth [27]	Predicted	Selfsup(S)	0.101	0.716	4.395	0.179	0.899	0.966	0.983
sfm-learner [4]	N/A	Selfsup(M)	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Vid2Depth [48]	N/A	Selfsup(M)	0.163	1.240	6.221	0.250	0.762	0.916	0.968
Geonet [5]	N/A	Selfsup(M)	0.155	1.296	5.857	0.233	0.793	0.931	0.973
SAVO [49]	N/A	Selfsup(M)	0.150	1.127	5.564	0.229	0.823	0.936	0.975
GANVO [50]	N/A	Selfsup(M)	0.150	1.141	5.448	0.216	0.808	0.939	0.975
SIGNet [24]	N/A	Selfsup(M)	0.133	0.905	5.181	0.208	0.825	0.947	0.981
DualNet [51]	N/A	Selfsup(M)	0.121	0.837	4.945	0.197	0.853	0.955	0.982
monodepth2 [7]	N/A	Selfsup(M)	0.115	0.913	4.873	0.193	0.877	0.959	0.981
Featdepth [25]	N/A	Selfsup(M)	0.104	0.729	4.481	0.179	0.893	0.965	0.984
PackNet-SfM [26]	N/A	Selfsup(M)	0.107	0.802	4.538	0.186	0.889	0.962	0.981
HR-Depth [27]	N/A	Selfsup(M)	0.104	0.727	4.410	0.179	0.894	0.966	0.984
SCSI [28]	N/A	Selfsup(M)	0.109	0.779	4.641	0.186	0.883	0.962	0.982
RM-Depth [32]	N/A	Selfsup(M)	0.108	0.710	4.513	0.183	0.884	0.964	0.983
selfVIO [13]	N/A	Selfsup(MI)	0.127	1.018	5.159	0.226	0.844	0.963	0.984
Proposed	Predicted	Selfsup(MI)	0.141	1.117	5.435	0.223	0.804	0.942	0.977

[a]Scale column represents whether the method predicts the real-world scale or not: 'Predicted' means the real-world scale is estimated and 'N/A' means the scale cannot be predicted, so the scale is taken from *the ground-truth depth* as in (33) for each frame to obtain the performance indices.

[b]Train column represents the training methods with the data: 'Unsup(S)' is the unsupervised method with stereo image pairs, 'Selfsup(S)' is the self-supervised method with sequences of stereo image pairs, 'Selfsup(M)' is the self-supervised method with sequences of monocular images, and 'Selfsup(MI)' is the self-supervised method with monocular image and the IMU measurement between consecutive images.

* The performance of other methods comes from each reference paper.

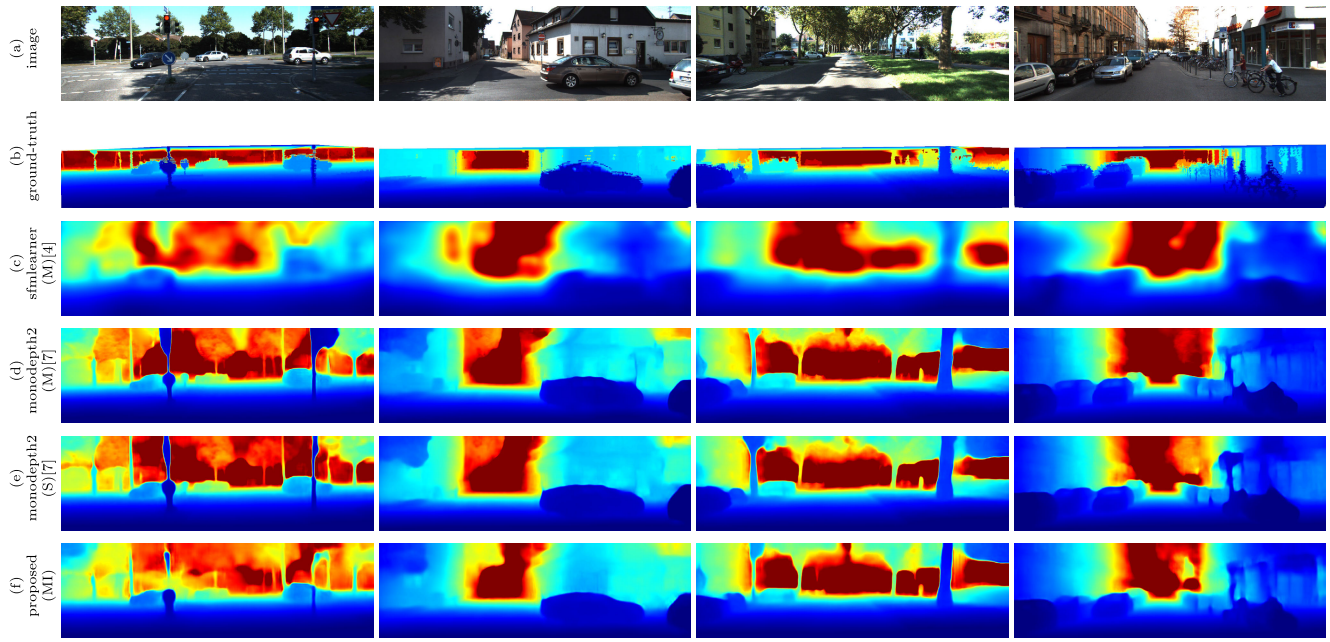


FIGURE 7. The depth prediction result on the Eigen split for qualitative comparison. The depth of ground-truth is generated from the lidar data with bilinear interpolation, and other methods are collected from the results provided by the authors of [4], [7]. (·) next to denotes the training data: (M) is monocular sequence, (S) is stereo sequence and (MI) is monocular sequence with IMU measurement. For the methods with (M), because no scale is estimated, the scale is taken from the *ground-truth depth* as in (33).

whole trajectory by the least square solution minimizing the translation part of RPE as

$$s_{\text{pose}} = \frac{\sum_i P_i \cdot \hat{P}_i}{\sum_i P_i \cdot P_i} \tag{38}$$

where $\cdot$ is a standard inner product in $\mathbb{R}^3$.

D. ABLATION STUDY

In the ablation study, we check the effectiveness of the proposed loss function. The first ablation study is intended to check whether the preintegration loss contributes to estimating a real-world scale. The second ablation study focuses on the bias regulation loss function described.

TABLE 3. Pose estimation performance in KITTI odometry dataset.

Methods	Scale[a]	Data[b]	Odometry 09				Odometry 10			
			ATE (m)	ATE (°)	RPE (m)	RPE (°)	ATE (m)	ATE (°)	RPE (m)	RPE (°)
[d]monodepth2 [7]	N/A	Selfsup(M)	147.750	21.2581	0.0821	0.0500	132.529	26.5673	0.0897	0.0527
[d]sfm-learner [4]	N/A	Selfsup(M)	136.811	24.5313	0.1496	0.0678	174.491	35.8786	0.1834	0.1038
[d]monodepth2 [7]	Predicted	Selfsup(S)	133.274	19.0951	**0.0701**	0.0528	149.966	30.4696	0.0747	0.0608
[d]FeatDepth [25]	Predicted	Selfsup(S)	72.149	10.6586	0.0655	0.0421	131.944	25.3016	0.0788	0.0586
[e]ORB-SLAM3 [8]	N/A	M[c]	**22.463**	**0.2975**	0.2781	**0.0202**	**20.040**	**2.7212**	**0.0462**	0.0607
[e]VINS-MONO [9]	Predicted	MI	30.142	0.9943	0.1251	0.0267	108.240	3.4176	0.1864	0.0704
proposed	Predicted	Selfsup(MI)	26.241	2.1733	0.1705	0.0380	63.664	6.9489	0.2493	**0.0427**

[a]Scale column represents whether the method predicts the real-world scale or not: 'Predicted' means that the real-world scale is estimated and 'N/A' means that the scale cannot be predicted, so the scale is taken from the *ground-truth pose* as in (38) across the trajectory.

[b]Data column represents the training method and utilized data: 'Selfsup(M)' is the self-supervised learning method with monocular sequences, 'Selfsup(MI)' is the self-supervised learning method with monocular sequences and the IMU measurements, 'Selfsup(S)' is the self-supervised learning method with stereo sequences, 'M' is classical monocular navigation, and 'MI' is classical monocular-inertial navigation.

[c]Due to the IMU initialization issue, monocular visual-inertial odometry failed. So instead, the monocular visual odometry is performed.

[d]The performance is obtained based on the weight provided by the authors of [4], [7], [25].
[e]The performance is obtained based on the code provided by the authors of [8], [9].

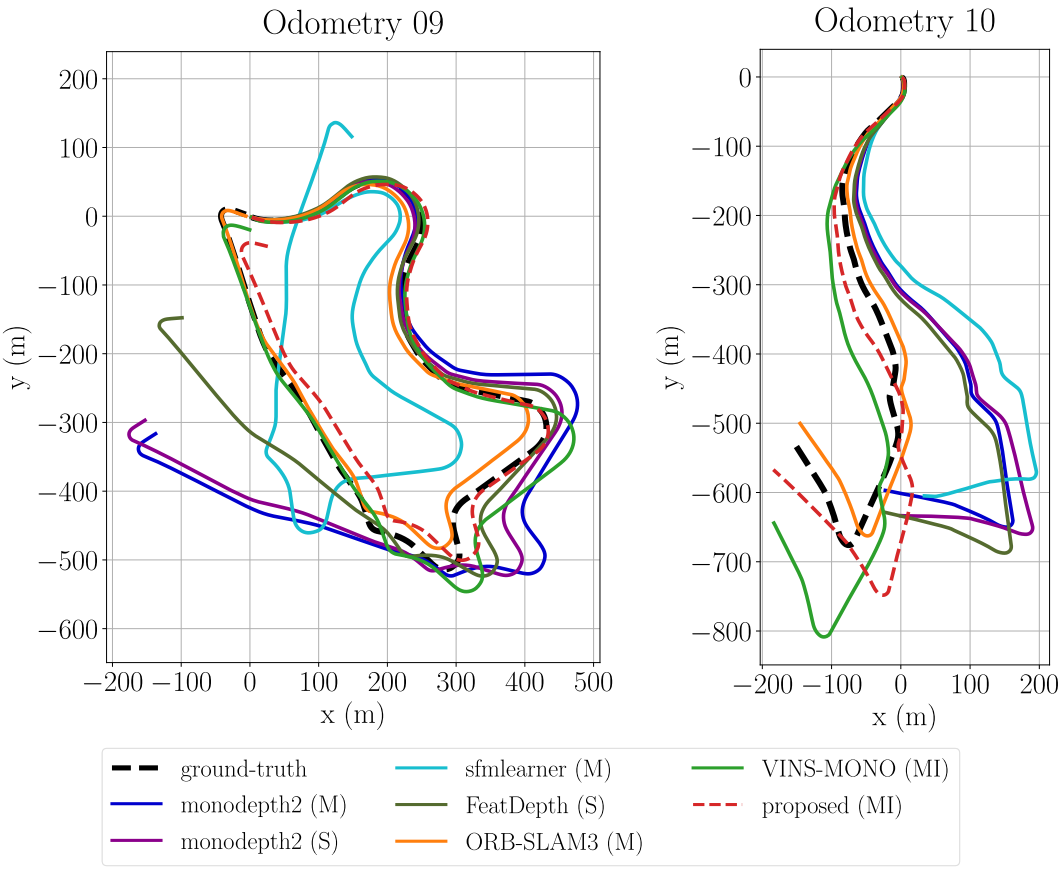


FIGURE 8. The top-down view of estimated trajectory of the KITTI odometry dataset. (·) next to the method denotes the training data for the deep-learning-based method or the operating data for the classical navigation: (S) is stereo sequence, (M) is monocular sequence and (MI) is monocular sequence with IMU measurements. For the methods with (M), because no scale is estimated, the scale is taken from the *ground-truth* as in (38). Each trajectory is aligned by fixing the initial point at origin.

1) ABLATION STUDY: PREINTEGRATION LOSS

In this ablation, we employ the proposed network architecture which receives both image and IMU as input, but turn off the preintegration loss function.

Table 1 shows the depth performance result. For no preintegration case, the relative depth seems good if the scale is taken from the ground-truth, but the raw depth is quite bad. On the other hand, the proposed method shows reasonable

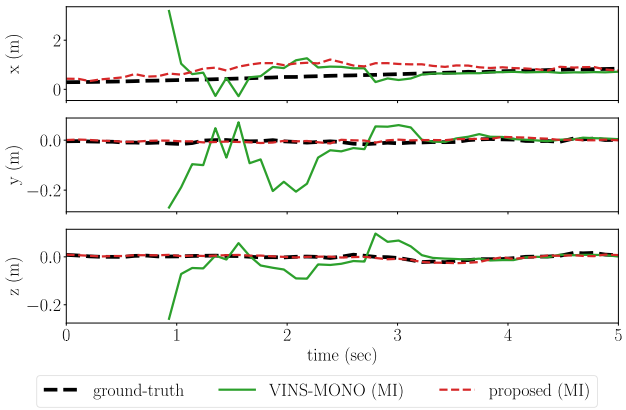


FIGURE 9. The relative translation prediction result in IMU coordinate at the beginning of the driving in the KITTI odometry 09 dataset.

(a) The experiment platform (b) The example of image

FIGURE 10. The setup of our experiment. Using vehicle (a), we experiment in an indoor environment like (b).

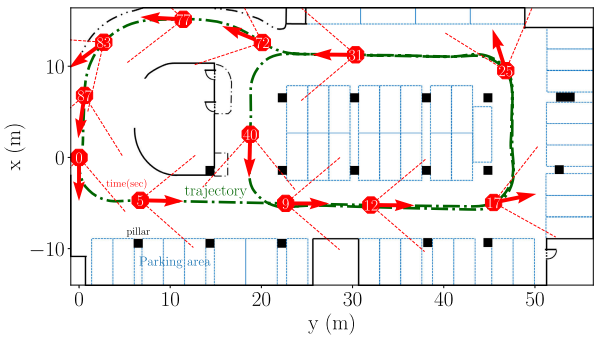


FIGURE 11. The overview of the experiment overlapped on the floor plan.

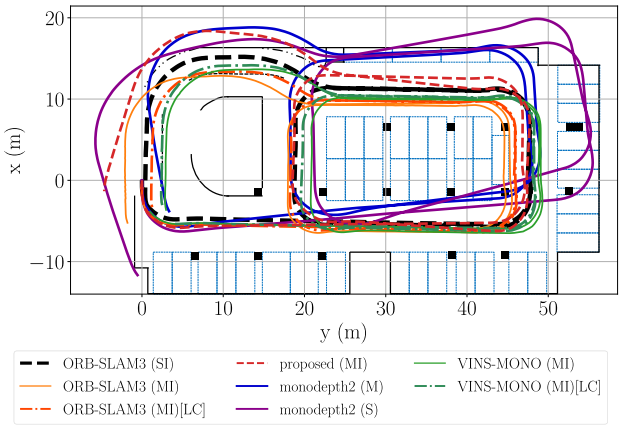


FIGURE 12. The top-down view of the predicted trajectories of the experiment. (·) next to the method denotes the training data for the deep-learning-based method or the operating data for the classical navigation: (SI) is stereo sequence with IMU measurements, (S) is stereo sequence, (MI) is monocular sequence with IMU measurements and (M) is monocular sequence. The method with [LC] has loop closing ability. For monodepth2 (M), because no scale is estimated, the scale is taken from *ORB-SLAM3 (SI)* as in (38). Each trajectory is aligned by fixing the initial point at origin.

performance even in the raw depth case. Fig. 3 is the scale prediction result of this ablation study, which shows that only the proposed method converges to the real-world scale value. We can conclude that without the preintegration loss, the relative depth can be trained due to the photometric consistency loss like self-supervised monocular methods, but the real-world scale is not learned.

2) ABLATION STUDY: BIAS REGULATION LOSS

In this ablation, we utilize the proposed network architecture which receives both image and IMU as inputs, but we turn off both bias regulation loss functions.

Fig. 4 shows the top-down view of the predicted trajectory depending on whether the bias regulation loss is on. Without the bias regulation loss, the scale prediction is wrong. As shown in Fig. 5, the magnitude of the bias is too large and the bias tends to follow the motion of the vehicle.

E. DEPTH PERFORMANCE VALIDATION

We validate the performance of depth prediction in the KITTI Eigen split. KITTI Eigen split is one of the famous data split methods, dividing the training and test data for the comparison of the learning-based depth prediction. It divides the dataset into the training data (33 videos containing 23,488 images) and the test data (697 images).

For quantitative analysis, we calculate the performance indices based on the ground-truth depth from Velodyne points for each test image. Since the Velodyne points provide sparse depth, we compare the pixels whose depth is available.

Table 2 shows the depth prediction performance of the proposed algorithm and state-of-the-art algorithm. The proposed method is less accurate than the state-of-the-art methods. However, it should be noted that the proposed method runs on monocular sequences with IMU measurements, which can be more easily collected than the stereo methods. Furthermore, the proposed method can predict the scale, which cannot be done by self-supervised monocular methods.

Fig. 6 shows the scale prediction result from the predicted depth. In this figure, the scale can be predicted by the proposed method and monodepth (S) that is the self-supervised stereo method. On the other hand, the self-supervised monocular methods, i.e., monodepth2 (M) and sfm-learner (M), cannot estimate the scale.

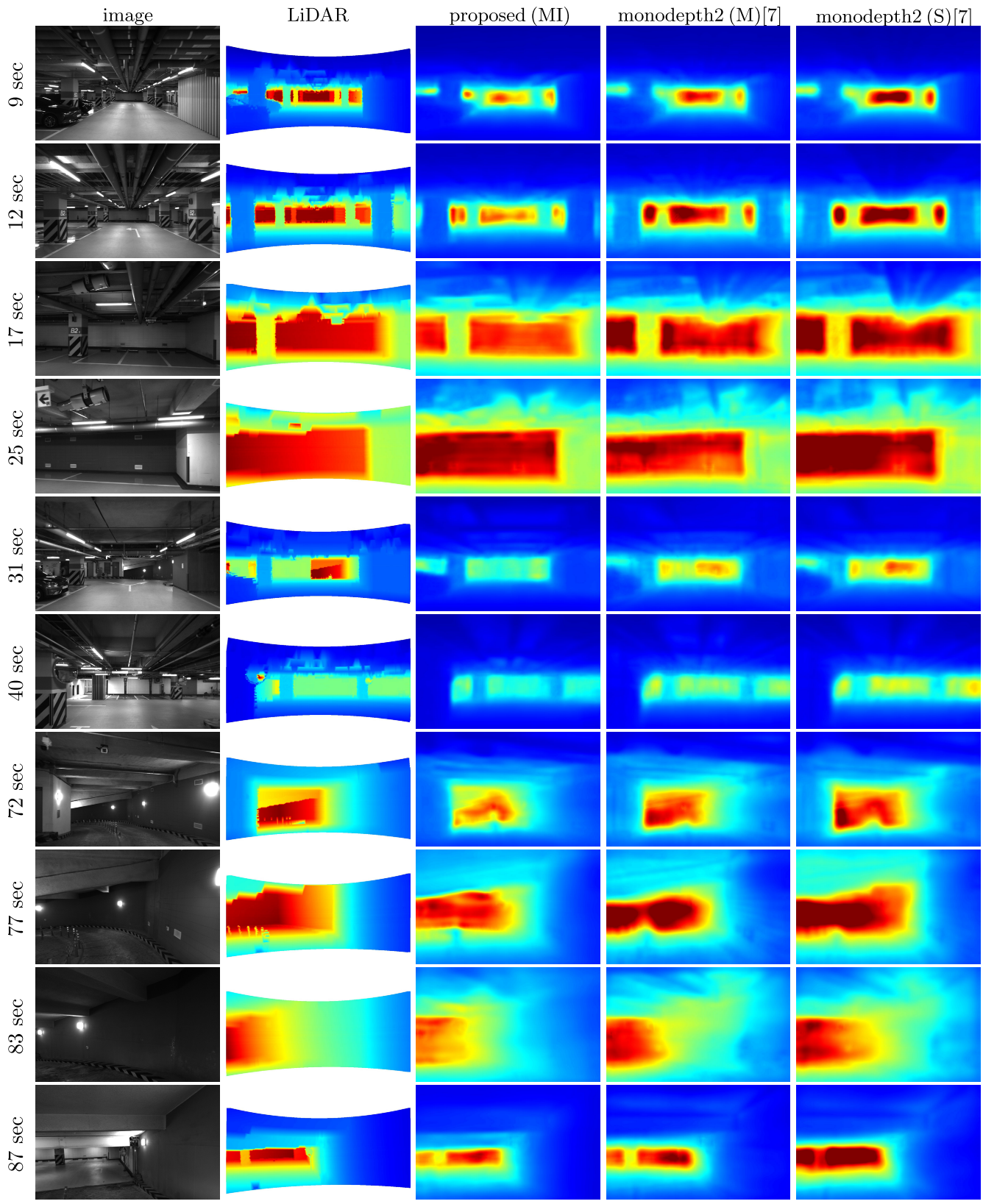


FIGURE 13. The depth prediction result of the experiment. The lidar data are interpolated using bilinear interpolation, and monodepth2 is trained by the authors' provided code. (·) next to denotes the training data: (S) is stereo sequence, (MI) is monocular sequence with IMU measurement and (M) is monocular image. For monodepth2 (M), because no scale is estimated, the scale is taken from *LiDAR* as in (33). The geometric information of each frame is expressed in Fig. 11.

Fig. 7 shows the depth map predicted from a single image. In this figure, the results of the proposed method, monodepth2 (M) and monodepth (S) correctly capture cars, trees, buildings, etc. In addition, they are qualitatively similar to the ground-truth color. Here, it should be noted that monodepth2 (M) and sfm-learner (M) yield no scale information, so the scale used in those methods is taken from the *grond-truth depth*. On the other hand, no ground-truth information is given to the proposed method and monodepth2 (S).

F. POSE PERFORMANCE VALIDATION

We test the pose prediction in the KITTI odometry dataset. The odometry 00-08 containing 19,600 images are used as the train data, and 09-10 containing 2,790 images are used as the test data. In this paper, odometry 03 is dropped since high-frequency IMU data is not provided.

In Table 3, the proposed method shows comparable performance to other state-of-the-art methods, considering that self-supervised monocular methods and classical monocular visual navigation methods cannot predict the scale, so the scale from the *ground-truth pose* was used.

Fig. 8 shows the top-down view of the predicted result. Since some classical methods provide no odometry information at the first step, we discard first few frames for fair comparison. The proposed method shows reasonable performance when compared with other methods, especially deep-learning-based methods.

When we focus on the beginning of the trajectory, VINS-MONO [9] shows poor performance before 5 seconds as in Fig. 9, perhaps due to the IMU initialization issue. For a car-driving case like the KITTI dataset, it is difficult to initialize IMU with a monocular camera because the motion of the vehicle is homogeneous. For this reason, the error of VINS-MONO is large in the beginning, and ORB-SLAM3 [8] with the monocular-inertial mode fails in this example.

G. EXPERIMENT

We experiment using a vehicle in an indoor environment as in Fig. 10. We employ a single monocular camera and IMU sensor during both the training and inference step for the proposed method. The vehicle is also equipped with a stereo camera, IMU and LiDAR sensors used for performance validation. In detail, two mvBlueCOUGAR cameras capture monotonic images at 10Hz to construct a stereo system, Lord Microstrain 3DM-GX3-25 attitude heading reference system (AHRS) provides angular velocity and acceleration data at 250 Hz and Velodyne VLP-32C Ultrapuck gives the point clouds of the surroundings at 10 Hz.

We drove the vehicle in the underground parking area of Seoul National University, building 39, and collected four sets of training data. Each dataset contains 1,000-1,600 images for approximately two minutes, with total 6,233 images for training and 870 images for inference. Fig. 11 shows the overview of the experiment. To evaluate the performance of the proposed method, we generate the depth map using the LiDAR measurement and the trajectory of the vehicle using stereo visual-inertial SLAM by ORB-SLAM3 [8]. The obtained trajectories are plotted onto the floor plan of the building for qualitative evaluation.

Fig. 13 shows the depth prediction result of the proposed method and monodepth2 provided by the authors of [7]. Since monodepth2 with monocular training denoted as monodepth2 (M) cannot predict the metric scale, we *take the metric scale* from the LiDAR measurement as in (33). The proposed method can predict geometric information such as pillars at 12-40 seconds, cars at 10 seconds, exits at 72-87 seconds and walls like monodepth2. The proposed method predicts the scale from monocular images and IMU measurements, but monodepth2 needs *stereo* images during the training step, otherwise it cannot predict the scale information.

Fig. 12 shows the pose prediction result of the proposed method, monodepth2 [7], VINS-MONO [9] and ORB-SLAM3 [8]. For VINS-MONO and ORB-SLAM3, we perform monocular visual-inertial navigation with and without a loop closing module. Additionally, we perform stereo visual-inertial navigation using ORB-SLAM3 to obtain an accurate trajectory of the vehicle. For monodepth2 with monocular training denoted as monodepth2 (M), the scale is taken from the stereo visual-inertial SLAM trajectory denoted as ORB-SLAM3 (SI) as in (38). The proposed method shows more accurate attitude estimation results than monodepth2 because by utilizing IMU sensor, which is commonly available in many platforms these days. Although the proposed method shows drifts at the end of the trajectory, it shows a competitive result compared with classical monocular visual-inertial navigation methods until the middle of the trajectory.

VI. CONCLUSION

In this paper, we propose self-supervised monocular depth estimation and odometry which addresses the scale ambiguity issue with raw IMU measurements. We design the loss function and network architecture to learn the scale information from IMU measurements. We show that the proposed method provides the estimated scale with comparable performance in the KITTI dataset and the additional experiment using an actual vehicle.

The proposed method adopts the state-of-the-art monocular self-supervised depth prediction and odometry and handles the IMU measurements to obtain scale-aware depth and pose, keeping the advantage of monocular self-supervision. Thus, the performance of the proposed method depends on that of the employed monocular depth estimation algorithm. We expect that the performance will be enhanced if the recent algorithm is adopted.

In addition, we propose seven additional loss functions, including the regularization loss functions, so the number of hyperparameters to balance the loss function terms increases from one to eight. Loss balancing is an important issue to achieve satisfactory performance. In this paper, we intuitively balanced the loss functions, but proper loss balancing can further increase the quality of the proposed method.

The proposed method, like self-supervised monocular methods, can train the network using the same type of data used for the inference, i.e., the proposed method can train from the data collected during the inference. This suggests the possibility to extend the proposed method to the online learning framework, in which the robot/vehicle learns the surrounding environments by itself during the inference step without additional device setup. This characteristic can help improve the estimation performance especially when the robot confronts new environments.

REFERENCES

[1] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2366–2374.

[2] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

[3] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. CVPR*, Jul. 2017, pp. 1851–1858.

[5] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[6] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5555–5564.

[7] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[9] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[10] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 1–7.

[11] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, "Selective sensor fusion for neural visual-inertial odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10542–10551.

[12] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6906–6913.

[13] Y. Almalioglu, M. Turan, M. R. U. Saputra, P. P. B. de Gusmão, A. Markham, and N. Trigoni, "SelfVIO: Self-supervised deep monocular visual–inertial odometry and depth estimation," *Neural Netw.*, vol. 150, pp. 119–136, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608022000752

[14] P. Wei, G. Hua, W. Huang, F. Meng, and H. Liu, "Unsupervised monocular visual-inertial odometry network," in *Proc. IJCAI*, Jul. 2020, pp. 2347–2354.

[15] E. J. Shamwell, K. Lindgren, S. Leung, and W. D. Nothwang, "Unsupervised deep visual-inertial odometry with online error correction for RGB-D imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2478–2493, Oct. 2020.

[16] K. Sartipi, T. Do, T. Ke, K. Vuong, and S. I. Roumeliotis, "Deep depth estimation from visual-inertial SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10038–10045.

[17] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1899–1906, Apr. 2020.

[18] J. Choi, D. Jung, Y. Lee, D. Kim, D. Manocha, and D. Lee, "SelfTune: Metrically scaled monocular depth estimation through self-supervised learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 6511–6518.

[19] L. Wang, J. Zhang, O. Wang, Z. Lin, and H. Lu, "SDC-depth: Semantic divide-and-conquer network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 541–550.

[20] J.-H. Lee and C.-S. Kim, "Monocular depth estimation using relative depth maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9729–9738.

[21] R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 740–756.

[22] 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. 270–279.

[23] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 155–163.

[24] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, and D. Bharadia, "SIGNet: Semantic instance aided unsupervised 3D geometry perception," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9810–9820.

[25] C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 572–588.

[26] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2485–2494.

[27] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan, "HR-Depth: High resolution self-supervised monocular depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 3, pp. 2294–2301.

[28] L. Wang, Y. Wang, L. Wang, Y. Zhan, Y. Wang, and H. Lu, "Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner?" in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12727–12736.

[29] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on CPU," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5848–5854.

[30] J. Liu, Q. Li, R. Cao, W. Tang, and G. Qiu, "MiniNet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, pp. 255–267, Aug. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620301544

[31] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 1–18.

[32] T.-W. Hui, "RM-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1675–1684.

[33] J. Gui, D. Gu, S. Wang, and H. Hu, "A review of visual inertial odometry from filtering and optimisation perspectives," *Adv. Robot.*, vol. 29, no. 20, pp. 1289–1301, 2015, doi: 10.1080/01691864.2015.1057616.

[34] C. Chen, H. Zhu, M. Li, and S. You, "A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives," *Robotics*, vol. 7, no. 3, p. 45, Aug. 2018. [Online]. Available: https://www.mdpi.com/2218-6581/7/3/45

[35] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9572–9582.

[36] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[37] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[38] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5391–5399.

[39] P. Kim, H. Lim, and H. J. Kim, "Visual inertial odometry with pentafocal geometric constraints," *Int. J. Control, Autom. Syst.*, vol. 16, no. 4, pp. 1962–1970, Aug. 2018.

[40] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015, doi: 10.1177/0278364914554813.
[41] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, 2015, pp. 2017–2025.
[42] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.
[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[45] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[46] 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, Sep. 2013.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[48] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5667–5675.
[49] S. Li, F. Xue, X. Wang, Z. Yan, and H. Zha, "Sequential adversarial learning for self-supervised deep visual odometry," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2851–2860.
[50] Y. Almalioglu, M. R. U. Saputra, P. P. B. D. Gusmao, A. Markham, and N. Trigoni, "GANVO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5474–5480.
[51] J. Zhou, Y. Wang, K. Qin, and W. Zeng, "Unsupervised high-resolution depth learning from videos with dual networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6872–6881.

CHUNGKEUN LEE (Graduate Student Member, IEEE) received the B.S. degree in mechanical and aerospace engineering from Seoul National University, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His research interests include robotic applications using a monocular camera with deep learning, including perception, navigation, and depth estimation.

CHANGHYEON KIM (Graduate Student Member, IEEE) received the B.S. and M.S. degrees from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in aerospace engineering. His research interests include 3-D reconstruction, visual navigation, and camera–IMU–LiDAR fusion.

PYOJIN KIM (Member, IEEE) received the B.S. degree in mechanical engineering from Yonsei University, in 2013, and the M.S. and Ph.D. degrees from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, in 2015 and 2019, respectively. He is currently an Assistant Professor with the Department of Mechanical Systems Engineering, Sookmyung Women's University, South Korea. Before joining Sookmyung Women's University, he was a Postdoctoral Researcher with Simon Fraser University, Canada. He was a Research Intern with Google (ARCore Tracking), Mountain View, in 2018. His research interests include indoor localization, 3-D computer vision, visual odometry, and visual SLAM for robotics.

HYEONBEOM LEE (Member, IEEE) received the B.S. degree in mechanical and control engineering from Handong Global University, in 2011, and the M.S. and Ph.D. degrees in mechanical and aerospace engineering from Seoul National University, in 2013 and 2017, respectively. From 2017 to 2018, he was a Senior Researcher with the Korea Institute of Machinery and Materials (KIMM). In September 2018, he joined the Department of Electronics Engineering, Kyungpook National University, Deagu, South Korea, as an Assistant Professor. His research interests include the autonomous navigation of aerial robots and mobile manipulators.

H. JIN KIM (Member, IEEE) received the B.S. degree from the Korea Advanced Institute of Technology (KAIST), in 1995, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Berkeley (UC Berkeley), in 1999 and 2001, respectively. From 2002 to 2004, she was a Postdoctoral Researcher of electrical engineering and computer science with UC Berkeley. In 2004, she joined the Department of Mechanical and Aerospace Engineering, Seoul National University, as an Assistant Professor, where she is currently a Professor. Her research interests include intelligent control of robotic systems and motion planning.