2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Unsupervised Learning of Depth and Pose Based on Monocular Camera and Inertial Measurement Unit (IMU)

Yanbo Wang, Hanwen Yang, Jianwei Cai, Guangming Wang, Jingchuan Wang, and Yi Huang

Abstract— **The main content of the research in this paper is the estimation of depth and pose based on monocular vision and Inertial Measurement Unit (IMU). The usual depth estimation network and pose estimation network require depth ground truth or pose ground truth as a supervised signal for training, while the depth ground truth and pose ground truth are hard to obtain, and monocular vision based depth estimation cannot predict absolute depth. In this paper, with the help of IMU, which is inexpensive and widely used, we can obtain angular velocity and acceleration information. Two new supervision signals are proposed and the calculation expressions are given. Among them, the model trained with acceleration constraint shows a good ability to estimate the absolute depth during the test. It can be considered that the model can estimate the absolute depth. We also derive the method of estimating the scale factor during the test from the acceleration constraint, and also achieve good results as the acceleration constraint does. In addition, this paper also studies the method of using IMU information as pose network input and as selecting conditions. Moreover, it analyzes and discusses the experimental results. At the same time, we also evaluate the effect of the pose estimation of the relevant models. This article starts by reviewing the achievements and deficiencies of the work in this field, combines the use of IMU, puts forward three new methods such as a new loss function, and conducts a test analysis and discussion of relevant indicators on the KITTI data set.**

I. INTRODUCTION

Even for a short period, people can infer their motion and the 3D structure of the scene. For example, in a street, people can easily locate obstacles and react quickly to avoid them. Years of geometric computer vision research have not been able to reproduce similar real-world modeling capabilities. Non-rigid objects, occlusions, and missing textures often occur in the real world. So why are humans so good at this task? One hypothesis is that humans build up a rich, structured perception of the world through past visual experiences, which are largely made up of our explorations, observations of large numbers of scenes, and models consistent with observations. From observations, people learn the laws of the world - roads are flat, buildings are vertical, cars are on the surface of the road, etc. People can apply this knowledge in new scenes even only with a monocular photo.

Typically, we mimic the way humans perceive the world by training a model that observes image sequences and predicts camera pose and scene structure. However, collecting a large amount of data with accurate depth ground truth for supervised training is a huge challenge. Instead, we perform unsupervised training using a large number of monocular video sequences that do not contain depth ground truth. At the same time, due to the limitations of monocular vision, we can only obtain relative depth information, which needs to be converted to absolute depth utilizing the true median alignment during testing. The IMU is a sensor that can obtain inertial information such as the acceleration and angular velocity of the device and has the advantages of low cost and the ability to obtain absolute scale, but it also has the disadvantage of excessive long-term cumulative error. Then, how to introduce the absolute scale information obtained by the IMU into the depth estimation network is the focus of this research.

The paper aims to study a joint unsupervised estimation network of depth and pose based on monocular vision and IMU, which can be trained using a large number of ground-truth-free monocular video sequences and matching IMU information. The corresponding absolute depth estimates are obtained from the target image and the corresponding IMU information. It is believed that the network can be applied in car positioning, unmanned driving, visual Simultaneous Localization and Mapping (SLAM), and other directions.

II. RELATED WORKS

A. Unsupervised Depth Estimation

In the case of missing depth ground truths, depth estimation models are generally trained using image reconstruction as a supervisory signal. The model takes a set of images as training input, estimates the depth of the given image and projects it onto adjacent views, and trains the model by minimizing the image reconstruction error. One form is self-supervised training from binocular pairs. Using epipolar geometric constraints, Godard et al. [1] propose an image reconstruction error based on the left-right consistency of binocular images to train an unsupervised depth estimation network. Zhan et al. [2] propose the reconstruction error based on depth features. And the reconstruction of the left and right views and the reconstruction between the two frames before and after are brought out. And Fabio et al. [3] propose a network named monoResMatch with a novel architecture.

*This work was supported in part by the Natural Science Foundation of China under Grant U1913204 and Pudong New Area Science Technology Development Fund (PKX2021-R01). The first two authors contributed equally. Corresponding Authors: Guangming Wang and Jingchuan Wang.

Y. Wang is with University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China. H. Yang is with School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. J. Cai is with Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China. G. Wang and J. Wang are with Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and Key Laboratory of System Control and Information Processing, Ministry of Education of China. Y. Huang is with Shanghai Weitong Vision Technology Co. , Ltd.

Another approach is to train the network using monocular video sequences. At this time, in addition to estimating the depth, the network also needs to estimate the pose change of the camera between frames, which is very difficult in the presence of moving objects. Estimating the camera pose only needs to be done during training to constrain the depth estimation network. Zhou et al. [4] are one of the first researchers to propose the use of monocular video sequences for deep unsupervised training. In addition, Yin et al. [5] propose GeoNet, an unsupervised learning framework that can estimate monocular depth, optical flow, and motion from videos. On the basis of Monodepth [1], Godard et al. [6] propose improvements such as minimizing reprojection error, full-resolution multi-scale sampling method, and automatic mask loss. In addition, Chen et al. [7], Reza et al. [8], and Wang et al. [9] also bring out new ideas on the topic. The unsupervised method is widely used in other works [10], [11], [12].

B. The estimation of depth and pose based on IMU

Similar to visual information, feature vectors can also be extracted from IMU information for deep network estimation of depth or pose. Clark et al. [13] propose a visual-inertial odometry network that learns pose estimates from visual and inertial sensor data using an end-to-end, sequence-to-sequence training method. Carlos et al. [14] propose to use an end-to-end machine learning method to train the odometry network. Chen et al. [15] propose an Inertial Odometry Net (IONet) that uses independent windows to segment inertial sensor data and uses a deep recurrent network to train and achieved good results in training and testing on their own datasets. Esfahani et al. [16] propose a three-channel deep inertial odometry network, AbolDeepIO based on the mathematical physical model of IMU for the positioning and navigation of unmanned vehicles. Shamwell et al. [17], [18] propose an unsupervised depth visual-inertial odometry based on Online Error Correction (OEC) for Red Green Blue Depth (RGBD) images. Han et al. [19] bring out DeepVIO, a self-supervised monocular visual-inertial odometry based on 3D geometric constraints, to obtain absolute trajectory estimates by combining planar optical flow features and IMU data. Fei et al. [20] propose a supervised visual depth estimation method, through the relationship between the surface normal vector of a specific class of objects and the direction of gravity, combined with the prior information of the inertial sensor, to obtain the depth estimation of the monocular camera. Wang et al. [21] also bring out ideas on 3D hierarchical refinement and augmentation unsupervised learning of depth and pose from monocular VideoG.

There are methods focusing on view composition, such as DeepStereo [22], Deep3D [23], and Appearance Flows [4]. More information can be obtained from previous methods such asResNet18 [24], ImageNet [25], FlowNetS [26], VIO [27]. Several methods are about optical flows [28], [29], [30], [31], [32], [33], [34].

We can see that monocular vision based depth estimation has the disadvantage of not being able to predict the absolute depth and requires a large amount of data with ground-truth depth and pose for training due to the supervised approach. Although IMU is commonly used in visual-inertial odometry, it is rarely used in in-depth estimation, and the absolute scale information of IMU information can help us get absolute depth estimation. Therefore, from this perspective, the paper uses IMU for the unsupervised monocular estimation of depth and pose, and has achieved certain results in absolute depth estimation.

III. THE UNSUPERVISED MONOCULAR ESTIMATION OF DEPTH AND POSE WITH IMU INFORMATION

This section mainly introduces the method of joint unsupervised estimation of depth and pose based on monocular vision and IMU according to the three lines of thought explained in the introduction:

1) Use the IMU information as a supervision signal, construct a suitable loss function to help the deep network learn the absolute depth estimation.
2) Use the IMU information as the input of the pose network to improve the accuracy of the pose network estimation, thereby improving the accuracy of the depth network estimation.
3) Use the IMU information as a screening condition to eliminate samples that do not meet the assumptions in the training set to improve the training effect.

A. Absolute Depth Estimation Based on IMU Information

1) Alignment of the True Value Median: Due to the limitations of monocular vision, only relative depth can be estimated when using monocular vision for depth estimation. When evaluating the depth estimation of monocular vision, Zhou [35] adopted the method of aligning the true value median to convert the relative depth to absolute depth, and then compared it with the true value of depth measured by liDAR. Zhou [35] calculated the error. The calculation is as follows:

$$\alpha = median(D_{gt})/median(D_{pred}). \tag{1}$$

For each image, the scale factor α is obtained by comparing the median of the true depth value D_{gt} with the median of the predicted depth D_{pred}, and then multiplying the scale factor into the depth map, which means converting the relative depth predicted by the deep network to the absolute depth. It can be used for error calculation and model evaluation. There are similar operations in monocular visual odometry. When evaluating the pose, the translation vector of the pose needs to be processed similarly to calculate the error.

2) Acceleration Constraint: In the previous section, we learn that monocular depth estimation needs to rely on the median alignment of the ground truth to convert the estimated depth value to absolute depth. In fact, if the effect of monocular vision depth estimation is good, the difference between the estimated depth map and the true depth is only a scale factor. At the same time, the acceleration information is included in the IMU information, which can be integrated twice to obtain a translation vector with an absolute scale. However, the quadratic integration method cannot be used

in the unsupervised depth estimation of monocular vision using only adjacent frames, because the magnitude of the initial velocity is not known. So we change our thinking and considered how to obtain the estimated acceleration through the estimated value in the network and compare it with the acceleration value of the IMU to extract the absolute scale information. In this way, the cumulative error effect of quadratic integration is eliminated.

To simplify the problem, we only consider a sub-sequence consisting of three monocular images: the target view (I_t) and one frame before and after the target view (I_{t-1} and I_{t+1}). Through the pose network, the pose transformation of the camera can be estimated from I_{t-1} to I_t and from I_t to I_{t+1}. The translation vectors are taken as $t_{t-1\to t}$ and $t_{t\to t+1}$. Since the camera shooting time interval $\Delta t \approx 0.1s$ in the KITTI dataset), it can be considered that the camera is approximately moving in a straight line with uniform acceleration. Then, the estimation of the acceleration vector $\hat{a}_t$ at the moment I_t can be calculated according to the relevant formula of uniform acceleration linear motion. Calculated as follows:

$$\overline{v}_{t-1\to t} = \frac{t_{t-1\to t}}{\Delta t}, \tag{2}$$

$$\overline{v}_{t\to t+1} = \frac{t_{t\to t+1}}{\Delta t}, \tag{3}$$

$$\hat{a}_t = \frac{\overline{v}_{t\to t+1} - \overline{v}_{t-1\to t}}{\Delta t}, \tag{4}$$

where $\overline{v}_{t-1\to t}$ is the average speed of the camera from I_{t-1} to I_t, and $\overline{v}_{t\to t+1}$ is the average speed of the camera from I_t to I_{t+1}.

After obtaining the estimated acceleration, we consider two ways to design the loss function. The first is to compare the estimated acceleration with the acceleration measured by the IMU, which is denoted as a_t in the formula, to obtain a scale factor so that the scale factor gradually approaches 1 so that the deep network can estimate the absolute depth. This approach follows naturally from the idea of ground truth median alignment. Another way is to directly make the estimated acceleration value close to the acceleration value of the IMU, which is common in supervised learning. In the selection of the loss function, we choose the L1 loss and Root Mean Squared Error (RMSE) loss commonly used in machine learning, which are represented by $loss\_fn()$ in the formula. We call this constraint the acceleration constraint, and the constructed loss function is called the acceleration loss, denoted as L_{acc}, and the calculation formula is as follows:

$$\alpha = \hat{a}_t/(a_t + \beta), \tag{5}$$

$$L_{acc} = loss\_fn(\alpha, 1), \tag{6}$$

$$L_{acc} = loss\_fn(\hat{a}_t, a_t), \tag{7}$$

where $\beta = 1e^{-7}$ is used to prevent the division by 0 error caused by the acceleration of 0 in the IMU data, and 1 is a unit vector.

Since the acceleration has three directions, we guess that the acceleration in the z-direction that is consistent with the direction of gravity may fluctuate greatly due to road bumps, which is not suitable for calculating errors. Therefore, in the subsequent experiments, we test the effects when adopting acceleration constraint only in the x direction and only in the z-direction. The effect of the acceleration constraint. In addition, to avoid the influence of accidental errors caused by the acceleration measured by the IMU, we adopt the method of calculating the average value of the acceleration of the three frames of the IMU as the true value of the acceleration. After obtaining the original data set, where the sampling frequencies are shown below, five frames which last 0.1s in total were used before and after I_t to find the IMU average value as the acceleration true value to avoid accidental errors.

TABLE I
THE USED SAMPLING FREQUENCY.

Situations	Sampling frequency
IMU and picture used before	10Hz
The original IMU	100Hz
The original picture	10Hz

3) Angular Velocity Constraint: Although the angular velocity information in the IMU does not contain absolute scale information, we still test its impact on unsupervised depth estimation in monocular vision. The calculation method is also similar to the acceleration constraint:

$$\overline{\omega}_{t-1\to t} = \frac{\theta_{t-1\to t}}{\Delta t}, \tag{8}$$

$$\overline{\omega}_{t\to t+1} = \frac{\theta_{t\to t+1}}{\Delta t}, \tag{9}$$

$$\begin{aligned} L_{ang} = {} & loss\_fn(\overline{\omega}_{t-1\to t}, \frac{\omega_{t-1} + \omega_t}{2}) \\ & + loss\_fn(\overline{\omega}_{t\to t+1}, \frac{\omega_t + \omega_{t+1}}{2}), \end{aligned} \tag{10}$$

where ω_{t-1}, ω_t and ω_{t+1} are the angular velocity measured by the IMU at I_{t-1}, I_t and I_{t+1}. $\overline{\omega}_{t-1\to t}$ and $\overline{\omega}_{t\to t+1}$ are the angular velocity estimated by the rotation angle which is described by Euler angles and obtained by the pose network from I_{t-1} to I_t and from I_t to I_{t+1}. Since the angular velocity is included in the IMU information, no second differentiation is required to obtain the angular acceleration. With the average angular velocity obtained, we use the average value of the angular velocity at both ends as the true value of the angular velocity to calculate the loss. In addition, since the rotation of the camera is almost always around the z-axis direction, which is the same as the direction of gravity, the effect of the angular velocity constraint that only considers the z-axis angular velocity is also tested.

4) Post-Processing Method: In the unsupervised monocular estimation of depth and pose, the pose network and the deep network are independent at test time and can be tested separately. This is because the pose network has little effect on the results of the deep network estimation. Therefore, we consider that if the scale factor is obtained by using the pose network while testing the deep network according to the acceleration constraint method, the absolute depth estimate can be obtained by multiplying the scale factor by the estimated depth during testing. This method is carried out at test time. Thus it is called a post-processing method. Calculated as follows:

$$\overline{v}_{t-1\to t} = \frac{t_{t-1\to t}}{\Delta t}, \tag{11}$$

$$\overline{v}_{t\to t+1} = \frac{t_{t\to t+1}}{\Delta t}, \tag{12}$$

$$\overline{a}_t = \frac{\overline{v}_{t-1\to t} - \overline{v}_{t\to t+1}}{\Delta t}, \tag{13}$$

$$\alpha = \hat{a}_t/(a_t + \beta). \tag{14}$$

When using the scale factor specifically, we find that if it is multiplied correspondingly as the true median value is aligned, it will cause a great error because the acceleration constraint is weak after all. The scale factor is only an estimated value. Therefore, we adopt the method of selecting the median of the scale factor as the scale factor of all estimated depths to avoid the influence of extreme values on the depth estimation. At the same time, we also test several different ways to obtain the scale factor, which will be explained in the following experiments.

B. Pose Network with IMU Information as Input

We believe that the inertial features contained in the IMU information are conducive to improving the estimation effect of the pose network, so we consider using the IMU information as the input of the pose network to improve the accuracy of the pose estimation. Thereby it indirectly improves the accuracy of the depth estimation. The sampling frequency of IMU is usually higher than that of visual information. Thus, for the same period such as between two consecutive images, IMU has a larger number of time series samples, which is very suitable for feature extraction using the LSTM network. Since the KITTI dataset provides two versions of the data, one has been synchronized with the sampling frequency of IMU data and visual data adjusted to 10Hz. The other has not been processed and maintains the original sampling frequency, where the time interval for visual data is 0.1s, and the time interval for IMU data is 0.01s. We design two different types of networks for these two cases.

For the 10Hz synchronized IMU information, we directly splice the feature vector output from the original pose network with the IMU data with accelerations in three directions and angular velocity in three axes, and send it to the fully connected layer to obtain the final pose estimate. For the original IMU information of 100Hz, we choose a bidirectional LSTM layer for feature extraction. After obtaining the inertial feature vector, it can be directly spliced with the visual feature vector to obtain the fusion vector. Finally, the fusion vector is sent to the fully connected layer to obtain the estimated pose.

C. Filters Based on IMU Information

As shown in Figure 1, the green coordinate system in the figure is the coordinate system referenced by the IMU to measure the angular velocity and acceleration, where the z-axis is parallel to the direction of gravity.

The IMU information includes angular velocity information and acceleration information, which can clearly understand the current motion state of the camera. From background work we know view synthesis methods are based on a fundamental assumption: the camera is moving and the scene remains stationary. Then, during the training process, we can use the IMU information as a screening condition to remove the samples that do not meet the requirements to prevent them from affecting the accuracy of the model. When selecting the filter conditions, we consider that if the camera keeps moving at a constant speed, the acceleration will be close to 0, which is the same as the case where the camera remains stationary. Moreover, the camera motion cannot be effectively filtered out. Therefore, we select more accurate angular velocity information to make judgments. In addition, since the rotational motion of the camera is almost all around the z-axis (parallel to the direction of gravity), we finally choose to use the z-axis angular velocity for judgment.

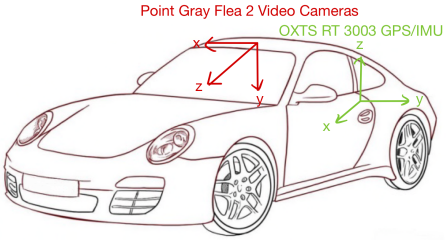

Fig. 1. Definition of each axis direction of the KITTI dataset.

Fig. 2. Example of still camera.

We select two representative subsequences in the dataset for analysis. The first is the case where the camera remains stationary, as shown in Figure 2. The angular velocity measurement value of analyzing IMU is in the order of $1e^{-4}rad/s$ or even lower, so we choose $1e^{-2}rad/s$ as the threshold during screening to judge whether the camera remains relatively stationary. It can be formulated as follows:

$$\mu = [ang_z > 0.01], \tag{15}$$

where [] is the Iverson bracket which is a square bracket notation that is 1 if the conditions inside the brackets are met, and 0 if they are not. ang_z means the angular velocity of the z-axis.

In the section on Acceleration Constraint, we mention that the acceleration constraint is based on the condition that the camera moves in a straight line with uniform acceleration. This assumption is not satisfied when the camera rotates at a large angle. Therefore, we consider using the angular velocity information measured by the IMU to rule out this situation. So in the dataset, we find the subsequences shown in Figure 3. The z-axis angular velocity measured by the IMU in this sequence is all above 0.1rad/s. According to the time estimation, the angular velocity of the z-axis is

Fig. 3. Example of large angle rotation.

90 degrees within 2.5s. The average angular velocity is $\overline{w} = \frac{\pi}{2} \div 2.5 \approx 0.628 rad/s$. Therefore, we select 0.1rad/s as the threshold value when implementing the acceleration constraint, and samples exceeding this threshold value will be excluded from the calculation of acceleration error. Similarly, when obtaining the scale factor during testing, we also need to do the corresponding screening to ensure that the scale factor satisfies the assumption of uniform acceleration linear motion.

IV. EXPERIMENTS

In this section, we first introduce KITTI dataset and the index we evaluate our depth and position evaluation algorithm on. Then we provide details about the training process of the experiment. The main part introduces the results of the experiment in the order of the previous chapter and gives the corresponding analysis.

A. Introduction to Datasets

All experiments in this paper are conducted on the KITTI data set. The KITTI data set contains data sets in many fields, such as depth, odometry, optical flow, etc. It is mainly used in the research of computer vision and automatic driving. We use Monodepth2 and SfMLearner to divide the data set. There are 39810 samples in the training set, 4424 samples in the verification set, and 694 samples in the test set of depth estimation. When evaluating the depth estimation, the sparse depth map collected by the liDAR will be converted into a dense depth map as the true value by bilinear interpolation. The KITTI odometer data set is used for the pose estimation evaluation. This is a subset of the KITTI dataset and contains 11 sequences with pose truth values that are computed by both GPS and IMU. In the test, we use the 9th sequence and the 10th sequence as the test set, and the rest sequences as the training set.

B. Experimental Details

All the models in this paper are based on the network framework of Monodepth2. Monodepth2 itself is an unsupervised depth and pose estimation model based on monocular vision. The unsupervised depth and pose estimation model based on IMU and monocular vision studied in this paper needs to enable the original model to read IMU data. In the introduction of the data set, we mention that the KITTI data set has two types: synchronous data set and original data set. If it is not indicated in the following experimental results, the KITTI synchronous data set with 10 Hz sampling of IMU and camera is used by default. If a dataset of raw 100 Hz IMU data is used, 100 Hz is indicated. The loss function is referred to as the L1 loss function if not specified, and the RMSE is indicated if the root mean square loss function is used. Without explanation, the default is to train for 20 rounds. Different experimental projects may have other notations, which will be explained.

C. Experiment of Absolute Depth Estimation Based on IMU Information

The acceleration constraint refers to the IMU measurement used as the acceleration true value. The acceleration constraint uses difference method xyz-m3, which means taking the average of the IMU measurements for the I_{t-1}, I_t, and I_{t+1} frames as the true acceleration. Acceleration constraint refers to the method of taking the absolute value of x-direction acceleration and y-direction acceleration and adding them to calculate the proportion. Acceleration constraint (difference mode XYZ) means that the acceleration in the three XYZ directions is considered in the error calculation, and the loss value is obtained by averaging in each direction and each sample. Acceleration Constraint (difference mode x) and Acceleration Constraint (difference mode z) refer to the calculation of acceleration losses using only the X and Z directions, respectively. Acceleration constraint (difference mode xyz-100Hz) means that the IMU data of the previous and next five frames are averaged as the true value of acceleration now. Since the original IMU data of 100Hz is used, this averaging method can reduce the impact caused by the accidental error of IMU measurement acceleration. And because the selected time span is only 0.1 s, which is shorter than the 0.2 s time span selected by the acceleration constraint (difference method xyz-m3), it should theoretically be more accurate than the method of selecting the average of IMU measurements of 1, and + 1 frames.

D. Training Results

From Table II we can find that the Monodepth2 network actually does not have the ability to estimate the absolute depth. Even in the 50th round of training, the error index and accuracy index are poor. The acceleration loss function is added to the network loss meter. After the calculation, the effect of the acceleration loss function using the proportional method is even worse than that of Monodepth2, and the method using the average of three frames as the true value of acceleration is not improved. We speculate that it may be because the acceleration values obtained by the pose network are generally large, while the acceleration values measured by the IMU are generally small, and the proportion obtained after division will change considerably due to the accidental error of the IMU measurement, which is unfavorable for the network to learn relevant features. Therefore, the acceleration loss function calculated by the proportional method is not very effective. However, we also find that the method of calculating the acceleration loss by using the ratio of the sum of the absolute values of the x-direction acceleration and the y-direction acceleration is more effective than Monodepth2. We speculate that since the sum of the absolute values of the accelerations in the two axes is used, the scale factor is

TABLE II

ABLATION RESULTS FOR DEPTH ESTIMATION USING EIGEN ET AL. TEST SPLIT ON KITTI DATASET. **A-DEPTH**: ALL-SCALE DEPTH. **H-DEPTH**: HIGHEST-SCALE DEPTH. **A-POSE**: ALL POSES. **H-POSE**: HIGHEST-REFINED POSE. **C-POSE**: COARSE POSES, WHICH ARE THE POSES EXCEPT FOR THE HIGHEST-REFINED POSE. : LOSSES AND GRADIENT ARE CALCULATED WITH TWO SIDES. →: LOSSES ARE CALCULATED WITH TWO SIDES, BUT GRADIENTS ARE ONLY CALCULATED FOR THE RIGHT SIDE AND STOPPED FOR THE LEFT SIDE.

Name of the experiment	Error index (the smaller, the better)				Accuracy indicator (larger is better)		
	AbsRel	*SqRel*	*RMSE*	log*RMSE*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 (train 20 generation)	0.969	15.128	19.199	3.491	0.000	0.000	0.000
Monodepth2 (train 50 generation)	0.967	15.084	19.171	3.448	0.000	0.000	0.000
Acceleration constraint (proportional)	0.977	15.438	19.383	3.928	0.000	0.000	0.001
Acceleration constraint (Proportional method m3)	0.980	15.506	19.413	4.022	0.000	0.000	0.001
Acceleration constraint (Proportional method x + y)	0.921	14.182	18.633	2.745	0.003	0.007	0.011
Acceleration constraint (Difference method XYZ)	0.458	4.887	10.908	0.683	0.126	0.307	0.581
Acceleration constraint (Difference method xyz-RMSE)	0.474	5.552	11.547	0.704	0.116	0.300	0.562
Acceleration constraint (Difference method xyz training 50 generations)	**0.339**	4.548	**8.908**	**0.404**	**0.457**	**0.736**	**0.867**
Acceleration constraint (Difference method xyz-m3)	0.559	5.982	12.089	0.922	0.069	0.150	0.281
Acceleration constraint (Difference method xyz-100Hz)	0.540	5.736	11.974	0.878	0.073	0.166	0.317
Acceleration constraint (Difference method x)	0.777	10.7	16.689	1.668	0.008	0.019	0.037
Acceleration constraint (Difference method z)	0.459	**4.183**	10.437	0.686	0.054	0.218	0.575

mainly determined by the part with the larger absolute value of the acceleration, so for the same error, the effect is smaller than that of taking the average value after the acceleration in one direction is scaled. We focus on the acceleration loss function computed in a difference manner. From the table, we can see that the effect of the method using the differential acceleration constraint is generally better than that of Monodepth2 and the proportional acceleration constraint. Specifically, the method of calculating the acceleration loss by averaging the accelerations in three directions can reduce the root mean square error from 19. 199 of the original model to 10. 908, which is only 56. 8% of the original model, and it can be considered that the model basically has the ability to estimate the absolute depth. Compared with the L1 loss function, the RMSE loss function is slightly less effective, but it can also achieve the effect of reducing the root mean square error to 11.547. In order to explore whether this method can continue to improve the effect with training, we test the effect of 50 generations of training, and find that the root mean square error decreased from 10.908 to 8.908. Compared with the original model, which decreases from 19.199 to 19.171, it can be considered that this method has the potential to continue to improve the effect. Research on how to process IMU data to make it a more accurate true value of acceleration: By observing two lines of experimental results of acceleration constraint (difference mode xyz-m3) and acceleration constraint (difference mode xyz-100Hz), We can see that using the IMU data at 100 Hz is slightly better than averaging the IMU data at - 1, and + 1. As mentioned above, the 100Hz IMU can be used to select acceleration measurement data closer to the time, which not only avoids the impact of accidental errors of the IMU, but also prevents the problem of inaccurate acceleration measurement caused by too long time span.

E. Experiment limitation

From Table II, all models using the acceleration constraint showed worse result than the original model, which consequently shows that the acceleration constraint is a weak constraint and is not accurate enough. The addition of the acceleration loss function makes the model obtain the ability of estimating absolute depth. However, it impairs the original ability of the model to estimate the relative depth accurately.

F. Experiment conclusion

Based on the introduction of the method, the related experiments are completed, and the experimental results are analyzed and discussed. Firstly, the KITTI data set used in the experiment and the evaluation indexes used for depth estimation and pose estimation are briefly introduced, and the details of the experiment are explained. Next, the depth estimation is evaluated on the KITTI data set in the order of the previous section. From the evaluation, we can find that the acceleration constraint we designed enables the model to estimate the absolute depth. Although the effect is not comparable to that of the original model with true median depth alignment, the model has basically the ability to estimate the absolute depth.

V. CONCLUSIONS

This paper mainly studies the unsupervised joint estimation of depth based on monocular vision and IMU, using a series of monocular video sequences without true depth and the corresponding IMU data for unsupervised training to obtain a model that can estimate depth and pose. In the introduction, the research background and significance of this subject are briefly introduced, and the relative position of the research content in the whole monocular unsupervised depth and pose estimation field is explained.

REFERENCES

[1] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," *The International Conference on Computer Vision (ICCV)*, pp. 3828–3838, October 2019.

[2] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 270–279.

[3] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 9799–9809.

[4] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[5] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1983–1992.

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 270–279.

[7] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7063–7072.

[8] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 5667–5675.

[9] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5555–5564.

[10] G. Wang, H. Wang, Y. Liu, and W. Chen, "Unsupervised learning of monocular depth and ego-motion using multiple masks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4724–4730.

[11] G. Wang, C. Zhang, H. Wang, J. Wang, Y. Wang, and X. Wang, "Unsupervised learning of depth, optical flow and pose with occlusion from 3d geometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 308–320, 2020.

[12] G. Wang, J. Zhong, S. Zhao, W. Wu, Z. Liu, and H. Wang, "3d hierarchical refinement and augmentation for unsupervised learning of depth and pose from monocular video," *arXiv preprint arXiv:2112.03045*, 2021.

[13] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] C. Marquez Rodriguez-Peral and D. Pe na, "Analysis of the effect of sensors for end-to-end machine learning odometry," in *The European Conference on Computer Vision (ECCV) Workshops*, September 2018, pp. 0–0.

[15] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "Aboldeepio: A novel deep inertial odometry network for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[17] E. J. Shamwell, S. Leung, and W. D. Nothwang, "Vision-aided absolute trajectory estimation using an unsupervised deep network with online error correction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2524–2531.

[18] 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 transactions on pattern analysis and machine intelligence*, 2019.

[19] L. Han, Y. Lin, G. Du, and S. Lian, "Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints," *arXiv preprint arXiv:1906.11435*, 2019.

[20] X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1661–1668, 2019.

[21] G. Wang, J. Zhong, S. Zhao, W. Wu, Z. Liu, and H. Wang, "3d hierarchical refinement and augmentation for unsupervised learning of depth and pose from monocular video," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[22] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5515–5524.

[23] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 842–857.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[26] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015, pp. 2758–2766.

[27] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 542–10 551.

[28] G. Wang, X. Wu, Z. Liu, and H. Wang, "Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 910–15 919.

[29] G. Wang, Y. Hu, X. Wu, and H. Wang, "Residual 3-d scene flow learning with context-aware feature extraction," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.

[30] G. Wang, X. Wu, S. Jiang, Z. Liu, and H. Wang, "Efficient 3d deep lidar odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[31] G. Wang, C. Jiang, Z. Shen, Y. Miao, and H. Wang, "Sfgan: Unsupervised generative adversarial learning of 3d scene flow from the 3d scene self," *Advanced Intelligent Systems*, vol. 4, no. 4, p. 2100197, 2022.

[32] G. Wang, X. Tian, R. Ding, and H. Wang, "Unsupervised learning of 3d scene flow from monocular camera," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4325–4331.

[33] G. Wang, S. Ren, and H. Wang, "Unsupervised learning of optical flow

with non-occlusion from geometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20 850–20 859, 2022.

[34] G. Wang, Y. Hu, Z. Liu, Y. Zhou, M. Tomizuka, W. Zhan, and H. Wang, "What matters for 3d scene flow network," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 38–55.

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