Deep Monocular Visual Odometry for Ground Vehicle

XIANGWEI WANG, (Member, IEEE), AND HUI ZHANG, (Member, IEEE)
Department of Control Science and Engineering, Tongji University, Shanghai 201804, China
Corresponding author: Hui Zhang (1510456@tongji.edu.cn)
This work was supported in part by the China Scholarship Council under Grant 201706260065.

ABSTRACT Monocular visual odometry, with the ability to help robots to locate themselves in unexplored environments, has been a crucial research problem in robotics. Though the existed learning-based end-to-end methods can reduce engineering efforts such as accurate camera calibration and tedious case-by-case parameter tuning, the accuracy is still limited. One of the main reasons is that previous works aim to learn six-degrees-of-freedom motions despite the constrained motion of a ground vehicle by its mechanical structure and dynamics. To push the limit, we analyze the motion pattern of a ground vehicle and focus on learning two-degrees-of-freedom motions by proposed motion focusing and decoupling. The experiments on KITTI dataset show that the proposed motion focusing and decoupling approach can improve the visual odometry performance by reducing the relative pose error. Moreover, with the dimension reduction of the learning objective, our network is much lighter with only four convolution layers, which can quickly converge during the training stage and run in real-time at over 200 frames per second during the testing stage.

INDEX TERMS Visual odometry, machine learning, motion analysis.

I. INTRODUCTION

The daily life of human-beings is increasingly involved with mobile robots including autonomous ground vehicles (AGV), unmanned aerial vehicles (UAV), and service robots. Mobile robots are mandatory to locate themselves during navigating and carrying out tasks in a complex environment. In the case the environment is unexplored where neither global positioning system (GPS) nor the environment map is available, incremental methods such as visual odometry (VO) are crucial. Incremental approaches calculate the robot ego-motion (including translation and rotation motion) relative to its starting coordinate system incrementally, which is independent on a constructed map. Most VO solutions utilize geometry-based methods to estimate the ego-motion \((R, t)\) by minimizing the re-projection error \([1]\) (named feature-based methods) or the photometric error \([2]\) (named direct methods). However, these approaches require accurate sensor calibration and manual parameters tuning to work well in different environments \([3]\). Furthermore, the parameter tuning process is always engineer-in-the-loop and time-consuming.

To diminish the human efforts for parameter tuning, many research efforts have been put into learning-based end-to-end methods. Ego-motion estimation with learning-based methods is started by Roberts \(et\ al.\) \([3]\), who try to learn a mapping from optical flow to 2D motion with a K-Nearest Neighbors model. Many other pioneering methods also explore to model the mapping from optical flow to ego-motion \([4]–[7]\). Wang \(et\ al.\) \([8], [9]\) firstly propose an end-to-end model to map from raw image to ego-motion. Besides, Wang \(et\ al.\) also consider sequential information by using recurrent neural networks. To reduce the dependency of labeled data, Zhou \(et\ al.\) \([10]\) propose an unsupervised method to predict image depth together with ego-motion using two networks, then compute the re-projected image residual as the loss function. After that, many enhanced works are proposed by adding additional 3D geometric loss \([11]\), binocular loss \([12]\), deep feature reconstruction loss \([13]\), dynamic and optical flow loss \([14]\) or adversarial loss \([15]\). To achieve robustness of the system, Klodt \(et\ al.\) \([16]\) and Yang \(et\ al.\) \([17]\) further propose to estimate the uncertainty of estimated ego-motion and depth. Clement \(et\ al.\) \([18]\) attempt to improve illumination robustness with a image translation model.

However, the performances of end-to-end learning-based approaches are still not promising, we claim that one of
the fundamental problems is the limited available training datasets. Learning-based methods always rely on a huge and diverse dataset to train a well-performed model, such as Imagenet [19] for object detection and Cityscape [20] for semantic segmentation. However, for ground vehicle ego-motion estimation, the most remarkable datasets such as KITTI [21] and Robotcar [22] are still limited in both data amount and diversity. The self-supervised methods such as Zhou et al. [10] can diminish the dependency of ground truth motion, but they do not solve the training data problem, as they still rely on the image sequences. To cope with training dataset limitation, Slinko et al. [23] propose to generate a training set by random re-projection based on RGB-D frames. Besides, Wang et al. [24] collect a bigger dataset with complex motion patterns, diverse environments and challenging light conditions through simulation.

We explore to learn the visual odometry model with the limited KITTI dataset by simplifying the learning target. We ask and answer the question: How about we only learn the majority motion of the vehicle (as shown in Figure 1)? As the motion of a ground vehicle is constrained by its mechanical structure and dynamics, focusing on majority motion will not lead to much pose displacement, while it can simplify the learning problem and reduce the data amount requirement. Additionally, as the presence of noise, the observed minority motion is always with a low signal-noise ratio, so focusing on learning the major motion of ground vehicles can be a desirable alternative solution.

![Motivation: focusing on the majority motion of ground vehicle to simplify the learning target.](https://github.com/TimingSpace/DMVOGV)

The constrained motion model of ground vehicles has been widely adopted in geometry-based visual odometry methods. Many approaches utilize the fixed height of the mounted camera as an absolute reference to recover the monocular scale [25]–[29]. Scaramuzza et al. [30] propose 1-point-RANSAC for ego-motion estimation to improve the realtime performance based on motion constraint and Ackerman Steer Principle [31]. Choi et al. [32] consider the abrupt bumps or camera vibration and relax the planar assumption. Scaramuzza et al. [33] use a homography method to calculate the ego-motion with feature points on the ground provided by an omnidirectional camera.

In this paper, we firstly quantitatively evaluated the ground truth pose displacement caused by ignoring some insignificant motion dimensions (named motion focusing) and explored to minimize the pose displacement by considering the ground vehicle motion model (named motion decoupling). Moreover, we constructed light convolutional neural networks with only four convolutional layers to learn the significant motion of the ground vehicle and conducted experiments to show that motion focusing motion decoupling can improve the ego-motion estimation performance. The structure of the whole system is shown in Figure 2. The main contributions of this paper are:

1) By quantitatively evaluating the ground truth pose displacement caused by motion focusing, we found that the displacement is relatively small, which can experimentally prove the feasibility of the motion focusing;
2) We analyse the reason for unexpected x-axis translation, model the relationship between x-axis translation and y-axis rotation and utilize that to reduce pose displacement caused by motion decoupling;
3) We conduct a comparative experiment on KITTI dataset to show that the proposed motion focusing and decoupling can reduce the training time and improve the learning performance;
4) We construct light convolutional neural networks to model the major motion of ground vehicles, the model is light enough to be trained on GPU with about 2G memory, and run in real-time on CPU (over 200 frames per second). For the benefit of the community, we release our source code.\(^1\)

The rest of this paper is organized as follows: Section II describes our approach including motion focusing and training detail. Our approach is evaluated on the KITTI dataset [21] in Section III. We conclude this paper and discuss future work in Section IV.

II. PROPOSED METHOD

In this section, we firstly introduce the data processing methods including motion focusing and motion decoupling in Section II-A; then the network structure and training detail about the ground vehicle visual odometry model are described in Section II-B.

A. MOTION FOCUSING AND MOTION DECOUPLING

Motion focusing is a general idea to simplify the learning target by focusing on the majority motion, which is described in Section II-A.1 and the pose displacement caused by motion focus is evaluated in Section III-B.1. Motion decoupling is to reduce the pose displacement caused by motion focusing, which is described in Section II-A.2 and evaluated in Section III-B.2. The learning improvement induced by motion focusing and motion decoupling is evaluated in Section III-B.3.

\(^1\)https://github.com/TimingSpace/DMVOGV
1) MOTION FOCUSING
Motion focusing is to ignore the insignificant motion of the ground vehicle and focus on modeling the majority motion. We use the regular camera coordinate system as the reference system when decomposing the motion (as shown in Figure 1), which is a right-handed system and defined as follows: the origin is the optical center of the camera, the z-axis is defined as the forward optical axis, the x-axis is horizontal rightward and the y-axis is vertical downward. The rotation motions around the x-axis, y-axis, and z-axis are represented by Euler angle $\psi$, $\phi$, and $\theta$ respectively. The translation motions along different axes are denoted as $x$, $y$ and $z$ respectively.

We quantitatively evaluate the motion pattern of a ground vehicle. The evaluation is deployed on KITTI visual odometry dataset [21], which is a typical dataset for ground vehicle motion estimation. We calculate the variance of the translation motions and rotation motions about different axes in KITTI sequence 00 as shown in Figure 3a. (More variance visualizations are available in the supplement, which are similar to Figure 3a).

Figure 3a shows that the majority motion of the ground vehicle is z-axis translation and y-axis rotation, so we propose to simplify the motion estimation target by only focusing on the majority motion and we call this proposal as motion focusing. The pose displacement caused by motion focusing is evaluated in Section III-B.1.

2) MOTION DECOUPLING
However, there is still unignorable translation motion along the x-axis as shown in Figure 3a. By the analysis of Table 1, we can know that more drifted pose are caused while ignoring x-axis motion. Nevertheless, considering the dynamics constraint, a ground vehicle is not able to move along x-axis much, how do we get 10% x-axis translation? When looking deeper into the motion pattern of a ground vehicle, we found that the x-axis motion is resulted by the motion representation method:

\[
\begin{pmatrix}
R \\
0
\end{pmatrix} = \begin{pmatrix}
I \\
0
\end{pmatrix} \begin{pmatrix}
R \\
0
\end{pmatrix} \begin{pmatrix}
t \\
0
\end{pmatrix}
\]

where $I$ is a $3 \times 3$ identity matrix. In this representation, the translation motion $t$ is prior to the rotation motion $R$. 
The 1D histogram in Figure 4b shows the distribution of the ratio \( \theta \), which is defined as

$$
\theta = \arctan\left( \frac{x}{z} \right)
$$

(2)

where \( x \) and \( z \) represent the x-axis translation and z-axis translation respectively. It is verified when we visualize the x-axis translation and y-axis rotation as shown in Figure 3b, as the two motions are highly correlated.

As Figure 3b can only visualize the local correlation of a sub-sequence, we utilize two histograms in Figure 4 to visualize the global relationship of all y-axis rotation angle \( \alpha \) and translation angle \( \theta \) in all KITTI sequence 00-10. The 1d histogram in Figure 4b shows the distribution of the ratio \( \alpha/\theta \) and the 2d histogram in Figure 4a visualizes the joint distribution of \( \alpha \) and \( \theta \). Both of the two distributions show that the y-axis rotation angle \( \theta \) and translation angle \( \alpha \) are correlated.

So how to reformulate the motion representation to reduce the motion correlation? A naive way is to reformulate the translation as

$$
\begin{pmatrix}
R & t \\
0 & 1
\end{pmatrix} = \begin{pmatrix}
R' & 0 \\
0 & 1
\end{pmatrix} \begin{pmatrix}
1 & t' \\
0 & 1
\end{pmatrix}
$$

(3)

In this formulation, the vehicle’s rotation is prior to translation, so the translation motion is relative to the reference system after rotation motion and won’t be remapped. The relationship can be derived that \( R' = R, t' = R^{-1}t \).

However, as shown in Figure 5a, the rotation angle around y-axis \( \theta \) is not equal to the translation angle \( \alpha \) produced by the rotation. We need to find out the relationship \( \alpha = f(\theta) \) between \( \alpha \) and \( \theta \), and then we can use the translation angle \( \alpha \) to recover the vehicle motion by only keeping the y-axis rotation \( \theta \) and car forward translation \( z \), with (4)

$$
(x, z) = (z' (\sin(\theta)), \cos(\theta))
$$

(4)

In Figure 5a, marker A is the center of the back axle of the vehicle, and marker B is the location where the camera is mounted. The distance between \( A \) and \( B \) is denoted as \( l \) which represents the camera location. The estimated translation distance of the vehicle by visual odometry is the distance between \( B \) and \( B' \) which is denoted as \( z' \). We simplify Figure 5a to Figure 5b.

According to Ackerman Steer Principle [31], \( OA \perp AB \) and \( OA' \perp AB' \), so \( \phi = 0.5\beta = 0.5\theta \). In triangle \( CBB' \), according to the Law of Sines,

$$
\frac{\sin(\gamma)}{\sin(\beta)} = \frac{l - \frac{z}{\gamma} \cos(\frac{\theta}{2})}{z'}
$$

(5)

considering \( \theta \) is close to zero, it is approximated that \( \cos(\frac{\theta}{2}) \approx 1 \), and \( \frac{\gamma}{\beta} \approx \frac{\sin(\gamma)}{\sin(\beta)} \), so

$$
\frac{\gamma}{\beta} \approx \frac{l - \frac{z}{\gamma}}{z'}
$$

(6)

We denote \( d = |AC| \approx 0.5|AA'| = 0.5z \), according to the Law of Cosines in triangle \( CBB' \)

$$
z'^2 = (l + d)^2 + (l - d)^2 - 2(l + d)(l - d) \cos(\beta) = 2l^2 + 2d^2 - 2(l^2 - d^2) \cos(\beta) \approx 4d^2
$$

(7)

Thus \( z' \approx z \). Then the relationship between the translation angle \( \alpha \) and rotation angle \( \theta \) is obtained

$$
\alpha = 0.5\beta + 0.5 \theta
$$

(8)

We use the translation angle \( a \) to reconstruct a rotation matrix \( R_a \),

$$
R_a = \begin{pmatrix}
\cos(\alpha) & 0 & \sin(\alpha) \\
0 & 1 & 0 \\
-\sin(\alpha) & 0 & \cos(\alpha)
\end{pmatrix}
$$

(9)
The desired vehicle forward motion $\dot{z}'$ is the third element of $t'$. Till now, the planner motion of a ground vehicle can be represented by two variables: rotation angle $\theta$ and remapped forward motion $\dot{z}'$. We focus on learning the two-dimension motions to simplify the learning target. The model and learning details are introduced in the next section and the improvement induced by motion focusing and decoupling will be evaluated in Section III-B.3.

### B. MODEL AND TRAINING

We construct a light model to learn the major motion of the ground vehicle. As shown in Figure 6, the model is mainly constructed with convolutional layers, and each convolutional layer is followed by a group normalization layer [34] and ReLU (Rectified Linear Unit) layer except for the final one. The same with Zhou et al. [10], we use a global average pooling layer [35] instead of a fully connected layer as the final layer to reduce overfitting. We observed that the optical flow of images captured by a ground vehicle is mainly horizontal, especially when a vehicle is turning, as shown in Figure 7, so we utilize convolutional layers with non-square kernels to achieve a larger horizontal field of perception. Besides, we adopt dilated convolutional layers [36] to obtain a larger field of perception with less parameters.

The input of the model is stacked gray image pairs, we not only use consequence images to construct image pairs (frame interval is 0), but a random frame interval among $[-4, 4]$ is chosen for each sample, which can be served as a data augmentation process. The output is the corresponding majority camera motions represented by the $y$-axis rotation $\theta$ and remapped forward translation $\dot{z}'$. $\dot{z}'$ is obtained based on (10). L2 loss is used as supervision,

$$L_2 = \|\theta_g - \theta_p\|_2 + \|\dot{z}_g - \dot{z}_p\|_2$$

(11)

The subscript $\theta_p$ and $\dot{z}_p$ represent predicted results, $\theta_g$ and $\dot{z}_g$ represent the ground truth. We use ADAM [37] to optimize the model parameters, and the learning rate is set as 0.001 with linear decay after 50 epoch.

After the model is trained, the rotation angle $\theta$ and forward motion $\dot{z}'$ can be obtained from the output of the model taking the new image sequences as input. We first calculate the translation angle $\alpha$ with (8) and assume that the rotations about other axes are zero, and construct rotation matrix $R_\theta$ and $R_\alpha$, then vehicle translation vector $t_\alpha = R_\alpha(0, 0, \dot{z}')^T$ (this equation is equivalent to (4), which is called motion recovering). The full vehicle motion matrix is

$$T_i = \begin{pmatrix} R_\theta & t_\theta \\ 0 & 1 \end{pmatrix}$$

(12)

The vehicle pose is then accumulated by

$$P_i = P_{i-1}T_i$$

(13)

### III. EXPERIMENTAL RESULTS AND DISCUSSION

We conduct four experiments to evaluate the performance of the proposed method on the KITTI dataset [21]. First, we describe the evaluation dataset and experiment platform. Second, we detail the four implemented experiments: pose displacement evaluation, motion decouple performance, ego-motion estimation improvement and comparison with other methods. At last, we discuss and analyze the experimental results.

#### A. DATASET AND PLATFORM

The KITTI [21] benchmark provides 22 testing sequences, the first 11 of which are with ground truth pose for evaluation. RGB images, gray images, and lidar point clouds are provided in each testing sequence. We only utilize the monocular gray images with ground truth poses (sequence 00-10 of KITTI dataset) for training and testing our model. Four different training-evaluating splits are used to quantitatively evaluate the ego-motion estimation improvement by proposed motion focusing and decoupling, which is detailed in Section III-B.3. to compare with other relative methods, in Section III-B.4, we use KITTI 00-08 for training and 09-10 for evaluation, which is the same with other learning-based methods for a fair comparison. We use the averages of RPE (short for relative pose error) including relative rotation errors and relative translation errors [21] as the evaluation metric.

Our algorithm is implemented in Python based on PyTorch, which is a reliable deep learning framework with convenient Python interface. Our code is publicly available online. The algorithm is tested on a personal laptop with 16GB of RAM, Intel Core i7-7700 CPU @ 2.80GHz and Nvidia 1060 GPU with 6GB of graphic memory. The testing environment is Ubuntu 18.04 with CUDA 10.0 and Python 3.6.9. The model training requires only 2.0G GPU memory when batch size is set as 30 and the prediction frequency is above 200 FPS (frames per second) with only CPU in the testing period.

#### B. EXPERIMENTAL RESULTS

First, we evaluate the pose displacement caused by motion focusing via evaluating their RPE in Section III-B.1. Second, in Section III-B.2, we evaluate the mitigation of pose displacement after motion decoupling. Third, the ego-motion estimation improvement by the proposed motion focusing and decoupling is evaluated in Section III-B.3. Last, we compare our results with other learning-based and geometry-based methods in Section III-B.4.

1) **MOTION DISPLACEMENT BY MOTION FOCUSING**

To show the feasibility of the motion focusing, we quantitatively evaluate how much the pose will be drifted when ignoring the part of or all other insignificant motion dimensions. We reconstruct the pose after motion reduction, and utilize RPE [21] to evaluate the pose displacement. The average RPE of KITTI sequence 00-10 are recorded in Table 1.
In Table 1, the column and row header name represent the kept rotation axes and translation axes respectively. The NID is shorted for the number of ignored dimensions. When we only keep z-axis translation and y-axis rotation, the RPE of the reconstructed path is 2.20% and the NID is 4. We visualize some reconstructed paths after motion focusing in Figure 9, which shows that the reconstructed horizontal path is acceptable with little displacement. The displacement is mostly accumulated on the z-axis. The displacement on the z-axis depends on the running environment, as when there are ups and downs, the displacement will be larger (sequence 10 in Figure 9b) and the environment is almost flat, the displacement is less (sequence 07 in Figure 9a). More visualized reconstructed paths are available in the supplement.

We also visualize the RPEs in Figure 8 for better comprehension. The average RPEs with different numbers of ignored dimensions are plotted by a blue line. We divide the
average RPE (cost) by conducting dimensions number (gain) as visualized by the yellow line in Figure 8, the ratio can be served as a cost-gain index. The cost-gain ratio for only keeping z-axis translation and y-axis rotation is relatively small.

**2) POSE DISPLACEMENT IMPROVEMENT BY MOTION DECOUPLING**

To show the efficiency of the proposed motion decoupling, we quantitatively evaluate how much the pose displacement is reduced by motion decoupling. As shown in the first row of Table 1, ignoring x-axis translation will cause more pose displacement (from 2.06% to 2.20%). The objective of motion decoupling is to reduce the pose displacement when we ignoring x-axis translation and the approach is described in Section II-A.2.

According to (8), the relationship between translation angle $\alpha$ and rotation angle $\theta$ is linear. However, the slope of linear mapping is not fixed but relative to the unfixed forward motion $z$. To simplify the problem, we first use the fixed slope parameter to transform all forward motion and test the RPE with different ratios. The result is shown in Figure 10a. When the ratio is set as 1.7, we receive the minimum RPE. It can be interpreted that the average vehicle forward translation is about $1.7 - 0.5$ meters when the vehicle is with rotation. To understand the bar chart better in Figure 10a, we use different colors for specific bars. When the translation-rotation angle ratio is zero (the red bar in Figure 10a), the RPE is about 2.20, this is the error without motion decoupling. The cyan bar is corresponding to the equal translation angle and rotation angle, this is the performance of naive decouple as 3. The black bar corresponding represents that the translation angle is half of the rotation angle, it is true when the camera is mounted on the center of back axles, and this ratio was used by Scaramuzza et al. [30].

A fixed ratio will ignore the influence of vehicle forward distance, we also try to use a dynamic ratio. We test different camera locations $l$ to calculate the ratio with (8), then evaluate the reconstructed path RPE as shown in Figure 10b. When the camera location is set as 0.4m away from the back axle, the RPE error is minimum. The black bar in Figure 10b also represents that the translation angle $\alpha = 0.5\theta$ which is the same with the black bar in Figure 10a.

The comparison of dynamic decoupling and static decoupling is visualized in Figure 11. In Figure 11, the blue bars represent the RPE without motion decoupling, the yellow and green bars represent the RPE of static decoupling (the ratio is set as 1.7) and dynamic decoupling (camera location $l$ is set as 0.4m) respectively, and the red bars represent the RPE when we keep both the x-axis and z-axis translation motion. It can be found that both dynamic decoupling and static decoupling can reduce the RPE; the RPE of dynamic decoupling is lower than that of static decoupling and closer to the RPE when we keep both x-axis translation and z-axis translation.

**3) PERFORMANCE IMPROVEMENT BY MOTION FOCUSING AND DECOUPLING**

We experiment to examine the influence of motion focusing and decoupling. We use the same training data to train two kinds models: 1) MFM (motion focusing model), only the y-axis rotation and z-axis translation are learned; 2) AMM (all motion model), all six-degrees-of-freedom motions are learned. The RPE over the same testing set could be an index to show the improvement. The experiment is conducted over different training-testing data splits to avoid the contingency. We record the loss changing curve of the training set and
the RPE of the testing set. As shown in Figure 12, the MFM model converges faster than the AMM model for all training split.

| Training | Testing | Learn All Motion Trans (%) | Rot (deg/m) | Learn $R_2$, $t_2$ Rot (%) | Rot (deg/m) |
|----------|---------|-----------------------------|-------------|-----------------------------|-------------|
| 00 02 04 06 08 10 | 26.8 | 0.137 | 23.9 | 0.110 |
| 00 02 04 06 08 10 | 18.3 | 0.095 | 16.7 | 0.070 |
| 00 02 04 06 08 10 | 17.6 | 0.091 | 16.9 | 0.076 |
| 00 02 04 06 08 10 | 15.3 | 0.082 | 13.2 | 0.065 |

4) COMPARISON WITH OTHER METHODS
We compare our performance with other learning-based and geometry-based methods. Our model is trained on KITTI sequence 00-08 and test on KITTI 09 and 10 and the data split is the same with other CNN-based (CNN is short for Convolutional Neural Network) methods [10], [13], [14]. The testing RPEs are recorded in Table 3 and Table 4. Because the models of SfM-Learner [10] and GeoNet [14] are trained without absolute scale in a self-supervised way, their paths are aligned with the ground truth path before the evaluation. The models of Zhan et al. [13], DeepVO [8], and our method are trained with the absolute scale, so no alignment is required. The scales of ORB-SLAM monocular [1] and LIBVISO [38] are also aligned with the ground truth path.

As shown in Table 3 our method outperforms CNN-based methods (ego-motion model is mainly constructed by convolutional layers) [10], [13], [14] and competitive with CNN-RNN (RNN is short for Recurrent Neural Network) based method [8] which can optimize the pose using temporal information. Comparison with two popular traditional methods LIBVISO monocular [38] and ORB-SLAM monocular [1], we get better average translation performance.

C. DISCUSSION
In this section, we will conclude the result, analyze the performance, and state the limitation of the proposed method.

1) THE EFFICIENCY OF THE PROPOSED METHOD
From the experimental results above, it can be concluded in four aspects. 1) Motion focusing will not bring in much pose displacement. The average RPE is only about 2%, which means that the vehicle pose will be about 2m drifted after running 100m. The path visualization shows that the reconstructed path is acceptable. 2) Motion decoupling can reduce pose displacement. Motion decoupling utilizes the correlation of y-axis rotation and x-axis translation to reduce the reconstructed pose displacement. Dynamic decoupling outperformed static decoupling. For dynamic decoupling, the camera location is required. In the above experiment, the camera location is calculated based on ground truth data. In practice, it can also be measured directly if your data is collected by yourself. 3) Motion focusing and decoupling can improve ego-motion estimation performance in two aspects. Firstly, it reduces the training time, all the training experiments converged within 20 epochs, but the model for all motion converged after about 60 epochs, so the motion focusing model can reduce about 2/3 training time compared to all motion model. Besides, even though the training ground truth pose is a little drifted after motion focusing and motion decoupling, the testing performance of the MFM is better than AMM. 4) During comparison with geometry-based methods, it is found that the geometry-based methods are not robust, and obtain diverse performance on different sequences. Our method is more stable and robust with better average performance. Our method also outperforms
other CNN-based methods and competitive with DeepVO [8] which utilizes RNN to improve performance. We get better relative rotation performance but worse translation performance than DeepVO.

2) WHY MOTION FOCUSING AND DECOUPLING WORKS
The reason for the better performance of motion focusing and decoupling is threefold: firstly, the motion of a ground vehicle is constrained and has imbalanced distribution, the ignoring of insignificant motion could not cause large pose displacement which is proved in Section III-B.1 and this is the fundamental basis of motion focusing; secondly, the insignificant motions are too limited to have enough signal-noise ratio, so the model will be easily distracted by the noise when aiming to model them; thirdly, when we focusing on only two-dimensions motions, the training task becomes much simpler. A light model can be adopted and training data is relatively abundant. It is experimentally proved that increased training data amount does improve the testing performance as shown in Table 2.

3) THE LIMITATION
The proposed method can obtain better performance when the car has an approximate planar motion. The performance will decrease when there is significant x-axis rotation. As shown in Figure 11, the RPE errors of sequence 09 and 10 are relatively high, because in these two sequences the motions of the ground vehicle are not planar as shown in Figure 9. To deal with this limitation, a feasible approach is to adopt other sensors like IMU (inertial measurement unit) to estimate the x-axis rotation, which can be a complement of the estimated planar motion by the proposed model.

Besides, another limitation is that it is assumed that the camera is mounted flat and looking forward. If the camera has a pitch angle, the forward motion of the vehicle will be mapped into a z-axis motion and y-axis motion. In this circumstance, the camera pitch angle $\sigma$ should be calibrated before training. Then the translation motion should be transformed as

$$t_\sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\sigma) & -\sin(\sigma) \\ 0 & \sin(\sigma) & \cos(\sigma) \end{pmatrix} t$$

These limitations can also be addressed by utilizing a multi-model structure for visual odometry, where each sub-model only focuses on one dimension motion of a ground vehicle and the six-degree-of-freedom can be learned with six separated models and in this case, the influence of models weight sharing should be analyzed.

IV. CONCLUSION
In this paper, we formulated the motion of a ground vehicle as two-degree-of-freedom motions by proposed motion focusing and motion decoupling. The feasibility of motion focusing was experimentally proved by quantitatively pose displacement evaluation. We further reduced the pose displacement by proposed motion decoupling based on the rotation model of a ground vehicle. We constructed a light CNNs to model the two-degree-of-freedom motions, which can run in real time on CPU. We experimentally proved that the motion focusing and decoupling can improve the ego-motion estimation performance and reduce the converging time. Comparison with other methods on KITTI dataset shows that the performance of the proposed method is comparable if not better than other end-to-end visual odometry approaches and more robust than geometry-based methods.

ACKNOWLEDGMENT
The authors would like to thank H. Yu and S. Gao from Carnegie Mellon University for providing advices for review and editing.

REFERENCES
[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 3, pp. 611–625, Mar. 2018.

[3] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, “Memory-based learning for visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom., May 2008, pp. 47–52.

[4] V. Guizilini and F. Ramos, “Semi-parametric models for visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom., May 2012, pp. 3482–3489.

[5] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfaglia, “Exploring representation learning with CNNs for Frame-to-Frame ego-motion estimation,” IEEE Robot. Autom. Lett., vol. 1, no. 1, pp. 18–25, Jan. 2016.

[6] R. Li, S. Wang, Z. Long, and D. Gu, “UnDeepVO: Monocular visual odometry through unsupervised deep learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 2043–2050.

[7] S. Wang, R. Clark, H. Wen, and N. Trigoni, “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks,” Int. J. Robot. Res., vol. 37, no. 4–5, pp. 513–542, Apr. 2018.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017.

[9] 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., Jun. 2018, pp. 5667–5675.

[10] R. Li, S. Wang, Z. Long, and D. Gu, “UnDeepVO: Monocular visual odometry through unsupervised deep learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 7286–7291.

[11] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and J. M. Reid, “Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 340–349.

[12] 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.

[13] Y. Almaliegolu, M. R. U. Saputra, P. B. D. Gusmao, A. Markham, and N. Trigoni, “GANVO: Unsupervised monocular visual odometry and depth estimation with generative adversarial networks,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 5474–5480.

[14] M. Klotz and A. Vedaldi, “Supervising the new with the old: learning SFM from SfM,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 698–713.

[15] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, “D3 VO: Deep depth, pose deep and pose uncertainty for monocular visual odometry,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1281–1292.

[16] L. Clement and J. Kelly, “How to train a CAT: Learning canonical appearance transformations for direct visual localization under illumination change,” IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 2447–2454, Jul. 2018.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3213–3223.

[19] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3354–3361.

[20] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The Oxford RobotCar dataset,” Int. J. Robot. Res., vol. 36, no. 1, pp. 3–15, Jan. 2017.

[21] I. Slinko, A. Vorontsova, D. Zhakov, O. Barinova, and A. Konushin, “Training deep SLAM on single frames,” 2019, arXiv:1912.05405. [Online]. Available: http://arxiv.org/abs/1912.05405.

[22] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, “TartanAir: A dataset to push the limits of visual SLAM,” 2020, arXiv:2003.14338. [Online]. Available: http://arxiv.org/abs/2003.14338.