The Visual Computer
https://doi.org/10.1007/s00371-023-02840-y

ORIGINAL ARTICLE

Self-supervised learning of monocular 3D geometry understanding with two- and three-view geometric constraints

Xiaoliang Liu[1,2] · Furao Shen[1,3] · Jian Zhao[1,4] · Changhai Nie[1,2]

Accepted: 1 March 2023

Abstract
The 3D geometry understanding of dynamic scenes captured by moving cameras is one of the cornerstones of 3D scene understanding. Optical flow estimation, visual odometry, and depth estimation are the three most basic tasks in 3D geometry understanding. In this work, we present a unified framework for joint self-supervised learning of optical flow estimation, visual odometry, and depth estimation with two- and three-view geometric constraints. As we all know, visual odometry and depth estimation are more sensitive to dynamic objects, while optical flow estimation is more difficult to estimate the boundary area moved out of the image. To this end, we use estimated optical flow to help visual odometry and depth estimation process dynamic objects and use a rigid flow synthesized by the estimated pose and depth to help learn the optical flow of the area that moves out of the boundary due to camera motion. In order to further improve the consistency of cross-tasks, we introduce three-view geometric constraints and propose a three-view consistency loss. Finally, experiments on the KITTI data set show that our method can effectively improve the performance of the occluded boundary area and the dynamic object area. Moreover, our method achieves comparable or better performance than other monocular self-supervised state-of-the-art methods in these three subtasks.

Keywords 3D geometry understanding · Optical flow estimation · Visual odometry · Depth estimation · Self-supervised learning · Dynamic scenes

1 Introduction

The 3D geometry understanding of dynamic scenes captured by moving cameras is one of the cornerstones of 3D scene understanding. Depth sensing, visual odometry, and optical flow estimation all play essential roles in 3D geometry understanding. They are widely used in various fields, such as augmented reality, autonomous driving, 3D-reconstruction, and robotics. However, none of them is a simple problem in computer vision.

For monocular depth and ego motion estimation, traditional methods are usually based on correspondence search [1, 2] and multi-view geometric [3]. For optical flow estimation, traditional methods are usually based on the variational model, which relies on prior assumptions to define an energy function and then optimize it [4, 5]. However, traditional methods tend to be sensitive to camera parameters and fragile in challenging settings, such as featureless places, motion blurs, and lighting changes. To address this problem, many methods propose to use deep learning technology for supervised learning in 3D geometric structure understanding, such as monocular depth estimation [6–8], visual odometry [9, 10], and optical flow estimation [11–14]. Although existing supervised methods have demonstrated outstanding performance, they also necessitate vast volumes of labeled data, which can be difficult and expensive to obtain. Furthermore, the amount of available labeled data for supervised training is still limited.

To address the above problems, much recent work has

✉ Furao Shen
frshen@nju.edu.cn
Xiaoliang Liu
xiaoliang_liu@smail.nju.edu.cn
Jian Zhao
jianzhao@nju.edu.cn
Changhai Nie
changhainie@nju.edu.cn

1 National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
2 Department of Computer Science and Technology, Nanjing University, Nanjing, China
3 School of Artificial Intelligence, Nanjing University, Nanjing, China
4 School of Electronic Science and Engineering, Nanjing University, Nanjing, China

Published online: 15 April 2023

focused on self-supervised learning methods without labeled data, such as [15–19]. Compared with supervised learning, self-supervised learning is more complicated. In self-supervised learning, depth pose learning is usually assumed to be in a completely rigid scene, and it is challenging to handle dynamic objects by themselves. However, optical flow is good at handling dynamic objects in the scene. Although depth sensing, visual odometry, and optical flow estimation are different tasks, they are all essential in 3D scene understanding, and many practical applications need to handle them simultaneously. Just as humans prefer to learn by mixing some related tasks, we can also learn to solve these three tasks by joining them. On the one hand, we can use the estimated optical flow to handle moving objects in depth pose learning. On the other hand, the rigid optical flow synthesized by depth pose can improve the performance of optical flow, especially in the boundary occlusion area.

Based on the above observation, we propose a new jointly self-supervised learning framework for monocular depth estimation, monocular pose estimation, and optical flow estimation in an end-to-end fashion. We use geometric relations to combine these three tasks. Compared with two-view geometry, three-view geometry can provide more information. Unlike previous works [20–25] that only used two-view geometric relationships, we consider both the two-view and three-view geometric relationships to connect the three "islands." Finally, our experiments show the effectiveness of introducing the three-view relationship, and our method achieves comparable or better performance than previous methods in these three subtasks. In summary, our paper makes the following contributions:

- We present a new jointly self-supervised learning framework for monocular depth estimation, ego motion, and optical flow estimation in an end-to-end system.
- We exploit three-view geometric relationships and propose a three-view consistency loss. In experiments, we verified their effects on overall performance.
- Experiments on the KITTI dataset [26] show that our method can effectively deal with occluded boundary areas and dynamic object area, and achieves comparable or better performance with other monocular self-supervised state-of-the-art methods in these three subtasks.

2 Related works

Traditional visual odometry (VO) and visual simultaneous localization and mapping (VSLAM) approaches Traditional VO and VSLAM approaches, which include geometry-based methods, can be further divided into indirect and direct methods. The standard approach of indirect method or sparse feature-based methods is to extract the feature points or lines of the raw images, match the features using feature descriptors, recover camera motion using epipolar geometry, and finally refine the pose through reprojection error minimization. The representative indirect methods are VISO2 [27] and ORB-SLAM2 [28].

Unlike indirect methods, direct methods directly estimate the camera pose by minimizing the photometric error between image pixels. Since direct methods directly use the brightness information of pixels, the time for feature detection and descriptor calculation can be saved. Typical direct methods are LSD-SLAM [29] and DSO [30].

Supervised learning of depth estimation, VO, and optical flow estimation With recent development of deep learning, great progress has been made in many tasks of 3D geometry understanding, including depth estimation [6, 7, 31], VO [9, 10, 32–37], optical flow estimation [11–14].

For depth estimation, Eigen et al. [6] propose a supervised monocular depth estimation method, which introduces a fully convolutional network. Based on [6], Liu et al. [7] introduce Conditional Random Fields (CRFs) into depth estimation and define a CRFs loss function. Unlike [7], Laina et al. [31] propose a single architecture that is trained end-to-end and does not rely on post-processing techniques, such as CRFs or other additional refinement steps. In difference to previous approaches using convolutional neural networks (CNNs), DepthFormer [36] and BinsFormer [37] employ Transformer [38] to further improve the performance of depth estimation.

For VO, Kendall et al. [32] first propose PoseNet, which uses CNN to learn the 6-DoF pose of the camera. Then, Li et al. [33] extend it to RGB-D. The above two methods are mainly related to relocalization problems. Costante et al. [34] implemented a frame-to-frame VO system based on CNN for the preprocessed optical flow. Then, Wang et al. [9, 10] propose an end-to-end VO method based on RCNN, which can catch up with the best method based on traditional VO [27]. Currently, DeepVO [9] is also a representative method of supervised learning. The supervised VO method also includes Vinet [35], which is the visual inertial VO with deep learning.

For optical flow estimation, FlowNet [11] is a pioneer in supervised optical flow estimation. Following the FlowNet, FlowNet2 [12] designs a stacked encoder–decoder network, which stacks multiple sub-networks and uses image warping between each sub-network. In contrast with [11, 12], SpyNet [14] introduces a classical spatial-pyramid formulation for flow estimation, which uses warping operation and estimate flow fields at each pyramidal level with a coarse-to-fine strategy. PWC-Net [13] combines sophisticated conventional strategies such as pyramid, warping, and cost volume into network design. PWC-Net achieves state-of-the-art performance on KITTI [26, 39] and MPI Sintel [40].

Self-supervised/unsupervised learning of depth estimation and optical flow estimation Despite the great success

of supervised learning methods, labeling large datasets is not trivial but very expensive in real-world scenes. Hence, many self-supervised/unsupervised learning approaches have been proposed, such as depth estimation [15, 16, 41], optical flow estimation [17, 19, 42–44].

For depth estimation, Garg et al. [15] first propose a self-supervised encoder–decoder network for monocular depth estimation with binoculars for training, which uses photometric loss and smoothness loss. Based on [15], Godard et al. [16] additionally considered left–right consistency loss, which enforces consistency between left and right views disparities. Pilzer et al. [41] exploit knowledge distillation and cycle-inconsistency to improve performance further.

For optical flow estimation, Ahmadi et al. [42] first propose an unsupervised network for optical flow estimation, which uses photometric loss for training. Yu et al. [43] additionally introduce a standard robust smoothness loss to constrain the spatial correlation of flow fields. However, photometric loss and smoothness loss cannot work in occlusion regions. Therefore, [17, 19, 44] propose to improve the performance of unsupervised learning by processing occlusion regions.

Self-supervised/unsupervised multi-task learning of 3d geometric understanding The inspiration for multi-task learning [46] is that humans like to mix some related tasks to learn. Multi-task learning usually shows higher performance than single-task learning. Recently, many works attempt to exploit the correlation of depth, VO, and optical flow using multi-task learning, such as [20–22, 24, 25, 47].

Zhou [47] first proposes a joint self-supervised learning of depth and ego motion from monocular videos. However, depth pose learning is difficult to deal with dynamic objects. To handle non-rigidity and occlusions, GeoNet [20] additionally adds optical flow for joint learning, but its optical flow network is not independent, and the estimated optical flow can only be obtained after depth pose. Therefore, DF-Net [21] proposes to use three sub-networks to achieve jointly training of depth, camera pose estimation, and optical flow. CC [22] further introduced motion segmentation, which can separate the scene into moving objects and static background. Unlike works [20–22, 47] that use monocular videos to train, Wang et al. [24] and Liu et al. [25] use binocular data to jointly train the sub-network.

Unlike previous works [20–22, 24, 25], our method not only uses the two-view geometric relationship, but also uses the three-view geometric relationship to improve the consistency of cross-tasks.

3 Method

3.1 System overview

Our goal is to jointly train depth network, camera pose network, and optical flow network using unlabeled monocular videos. The overall framework of our method is shown in Fig. 1.

For two-view relationships, given two consecutive frames (I_{t1}, I_{t2}) sampled from an unlabeled video, we first estimate their depth maps (D_{t1}, D_{t2}) using the depth network and forward–backward 6-DoF relative poses $(T_{t1\to t2}, T_{t2\to t1})$ using the pose network. At the same time, we predict forward–backward optical flow fields $(F_{t1\to t2}, F_{t2\to t1})$ between them using the flow network. With the estimated depth map (D_{t1}) and 6-DoF relative pose $(T_{t1\to t2})$, we can produce the rigid flow $(F^{\text{rigid}}_{t1\to t2})$, which is the optical flow that is purely induced by the camera motion. Then, we can obtain the rigid region weight $(R_{t1\to t2})$ from the optical flow $(F_{t1\to t2})$ and the rigid flow $(F^{\text{rigid}}_{t1\to t2})$. The non-occluded region mask $(O_{t1\to t2})$ is estimated using forward–backward optical flow fields $(F_{t1\to t2}, F_{t2\to t1})$ as described in [44]. The auto mask $A_{t1\to t2}$ is proposed by [45]. In the next step, we can synthesize reference images $\widehat{I}_{t2}$ and $\widetilde{I}_{t2}$ by warping I_{t2} using $F^{\text{rigid}}_{t1\to t2}$ and $F_{t1\to t2}$.

For three-view relationships, previous works [20, 22, 47] consider input N-view ($N \geq 3$) image sequences during training, but they only use the two-view geometric relationships. As we all know, three-view geometric relationships are more informative and robust than two-view geometric relationships. To this end, we use the corresponding incidence relations of trifocal tensor [3] on point–point–point to propose a new cross-task consistency loss, three-view consistency loss.

Finally, we use brightness constancy and cross-task consistency for optimization. Similarly, we also use forward–backward consistency loss and smoothness loss for optimization. Our overall objective function can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{total}} = {} & \lambda^{\text{rigid}}_{\text{bc}}\mathcal{L}^{\text{rigid}}_{\text{bc}} + \lambda^{\text{flow}}_{\text{bc}}\mathcal{L}^{\text{flow}}_{\text{bc}} \\ & + \lambda^{\text{rigid}}_{\text{ct}}\mathcal{L}^{\text{rigid}}_{ct} + \lambda^{\text{tri}}_{ct}\mathcal{L}^{\text{tri}}_{\text{ct}} \\ & + \lambda^{\text{depth}}_{\text{fb}}\mathcal{L}^{\text{depth}}_{fb} + \lambda^{\text{pose}}_{fb}\mathcal{L}^{\text{pose}}_{\text{fb}} + \lambda^{\text{flow}}_{\text{fb}}\mathcal{L}^{\text{flow}}_{\text{fb}} \\ & + \lambda^{\text{depth}}_{s}\mathcal{L}^{\text{depth}}_{s} + \lambda^{\text{flow}}_{s}\mathcal{L}^{\text{flow}}_{s},\end{aligned} \tag{1}$$

where $\lambda^{(\cdot)}_{(\cdot)}$ are the weights for each term. Our total loss function $\mathcal{L}_{\text{total}}$ consists of four parts, as follows:

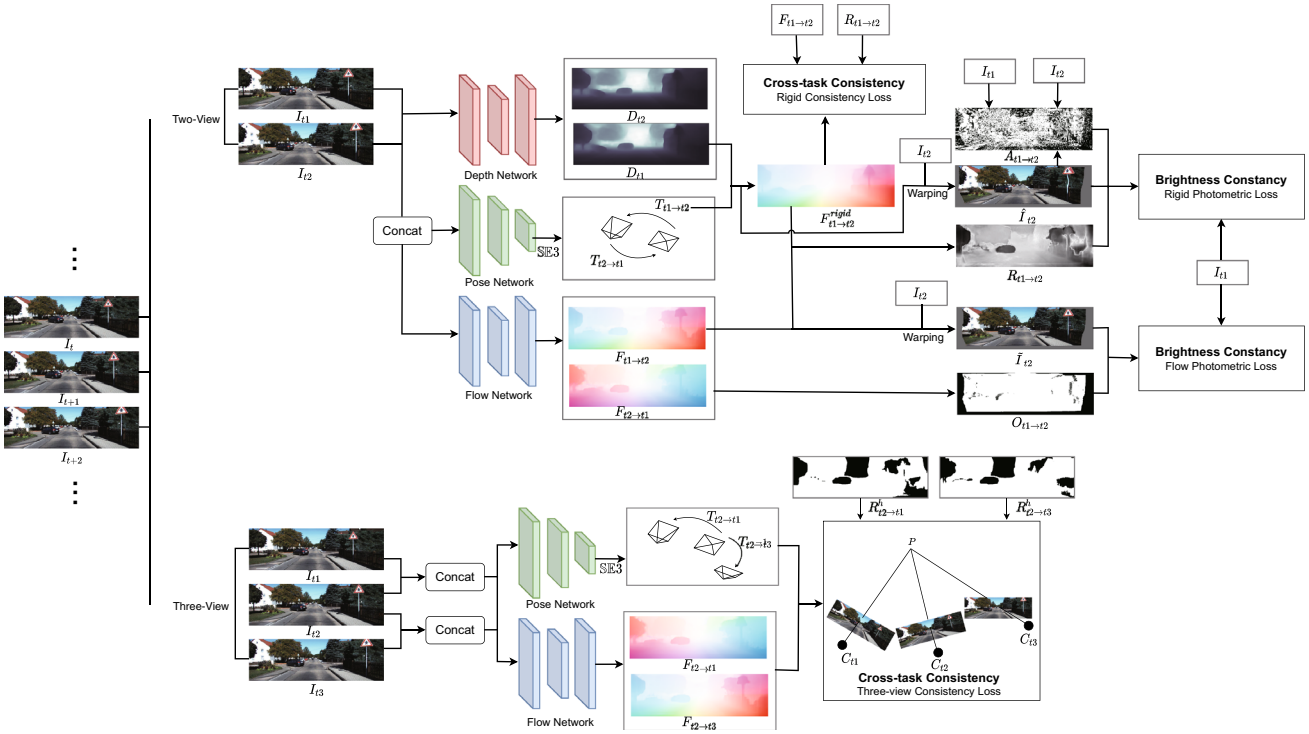


Fig. 1 Overview of our self-supervised joint learning framework. **a** Two view: given two consecutive frames (I_{t1}, I_{t2}) sampled from an unlabeled video, we first estimate their depth maps (D_{t1}, D_{t2}) using the depth network and forward–backward 6-DoF relative poses ($T_{t1\to t2}$, $T_{t2\to t1}$) using the pose network. At the same time, We predict forward–backward optical flow fields ($F_{t1\to t2}$, $F_{t2\to t1}$) between them using the flow the camera motion. Then, we can obtain the rigid region weight ($R_{t1\to t2}$) from the optical flow ($F_{t1\to t2}$) and the rigid flow ($F^{\text{rigid}}_{t1\to t2}$). The non-occluded region mask ($O_{t1\to t2}$) is estimated using forward–backward optical flow fields ($F_{t1\to t2}$,$F_{t2\to t1}$) as described in [44]. The auto mask $A_{t1\to t2}$ is proposed by [45]. In the next step, we can synthesize reference images $\widehat{I}_{t2}$ and $\widetilde{I}_{t2}$ by warping I_{t2} using $F^{\text{rigid}}_{t1\to t2}$ and $F_{t1\to t2}$. **b** Three-view: given three adjacent frames (I_{t1}, I_{t2}, I_{t3}), we can estimate the relative poses ($T_{t2\to t1}$, $T_{t2\to t3}$) and optical flows ($F_{t2\to t1}$, $F_{t2\to t3}$). Next, we can get the trifocal tensor from the relative poses ($T_{t2\to t1}$, $T_{t2\to t3}$) and use the optical flows ($F_{t2\to t1}$, $F_{t2\to t3}$) to match images. Finally, using the trifocal tensor incidence relation on point–point–point, we propose the three-view consistency loss

- Photometric loss, which includes rigid photometric loss $\mathcal{L}^{\text{rigid}}_{\text{bc}}$ and flow photometric loss $\mathcal{L}^{\text{flow}}_{\text{bc}}$, is the most critical signal for self-supervised depth pose and flow learning.
- Cross-task consistency loss, which includes rigid consistency loss $\mathcal{L}^{\text{rigid}}_{\text{ct}}$ and three-view consistency loss $\mathcal{L}^{\text{tri}}_{\text{ct}}$, provides mutual communication between the three tasks.
- Forward–backward consistency loss, which includes depth forward–backward consistency loss $\mathcal{L}^{\text{depth}}_{\text{fb}}$, pose forward–backward consistency loss $\mathcal{L}^{\text{pose}}_{\text{fb}}$, and flow forward–backward consistency loss $\mathcal{L}^{\text{flow}}_{\text{fb}}$, provides self-constraint for each task and can further improve the performance of each network.
- Smoothness loss, which includes depth smoothness loss $\mathcal{L}^{\text{depth}}_{s}$ and flow smoothness loss $\mathcal{L}^{\text{flow}}_{s}$, encourages the predicted depth and estimated optical flow to be smooth.

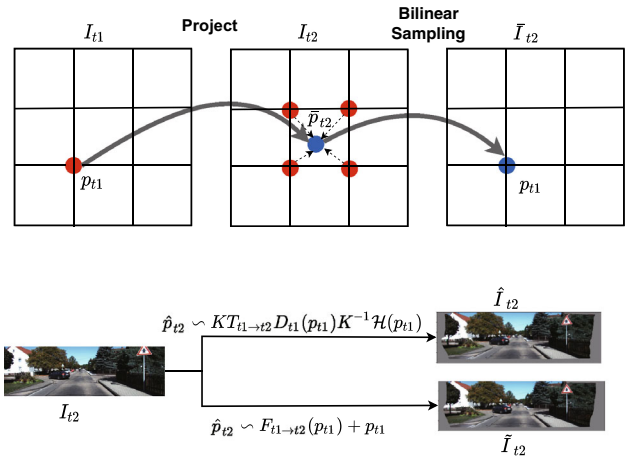


Fig. 2 Bilinear interpolation used in image reconstructed

3.2 Brightness constancy

3.2.1 Image reconstruction

The key idea of self-supervised methods is to utilize the photometric discrepancy between the reconstructed image $\bar{I}$ and the original image I. The image reconstruction in this work includes reconstruction based on predicted depth map (D) and relative pose (T), and reconstruction based on estimated optical flow (F).

Let p_{t1} denote the pixel coordinate of a pixel in frame I_{t1}. We can obtain its projected coordinates $\widehat{p}_{t2}$ and $\widetilde{p}_{t2}$ in frame I_{t2} by

$$\widehat{p}_{t2} \backsim K T_{t1\to t2} D_{t1}(p_{t1}) K^{-1} \mathcal{H}(p_{t1}), \tag{2}$$

$$\widetilde{p}_{t2} \backsim F_{t1\to t2}(p_{t1}) + p_{t1}, \tag{3}$$

where K denotes the camera intrinsic matrix, $\mathcal{H}(p_{t1})$ denotes the homogeneous coordinates of p_{t1}. Finally, as shown in Fig. 2, the reconstructed images $\widehat{I}_{t2}$ and $\widetilde{I}_{t2}$ can be obtained using bilinear interpolation.

3.2.2 Photometric loss

According to the reconstructed images $\widehat{I}_{t2}$ and $\widetilde{I}_{t2}$, we can obtain the *rigid photometric loss* ($\mathcal{L}_{\text{bc}}^{\text{rigid}}$) and the *flow photometric loss* ($\mathcal{L}_{\text{bc}}^{\text{flow}}$), respectively. Their definitions are as follows:

$$\mathcal{L}_{\text{bc}}^{\text{rigid}} = \frac{1}{|V_1|} \sum \rho(I_{t1}(p), \widehat{I}_{t2}(p)) V_1 R_{t1\to t2}, \tag{4}$$

$$\mathcal{L}_{\text{bc}}^{\text{flow}} = \frac{1}{|V_2|} \sum \rho(I_{t1}(p), \widetilde{I}_{t2}(p)) V_2 O_{t1\to t2}, \tag{5}$$

$$V_1 = V_{t1\to t2}^{\text{rigid}} A_{t1\to t2}, \; V_2 = V_{t1\to t2}^{\text{flow}} O_{t1\to t2}, \tag{6}$$

where $\widehat{I}_{t2}$ and $\widetilde{I}_{t2}$ are both reconstructed images by warping I_{t2}. $V_{t1\to t2}^{\text{rigid}}$ and $V_{t1\to t2}^{\text{flow}}$ are valid projection masks that are successfully projected from I_{t1} to I_{t2}. $A_{t1\to t2}$ is the auto mask that is proposed by [45]. [$\mathbb{1}$] is an Iverson bracket. $O_{t1\to t2}$ is the non-occluded region mask that is estimated using reverse option flow $F_{t2\to t1}$ as described in [44]. $R_{t1\to t2}$ is the rigid region weight, which is defined as follows

$$R_{t1\to t2} = 1 - \frac{1}{(H^2+W^2)^{1/2}} \| F_{t1\to t2} - F_{t1\to t2}^{\text{rigid}} \| \tag{7}$$

$$F_{t1\to t2}^{\text{rigid}}(p_t) = \widehat{p}_{t2} - p_{t1}. \tag{8}$$

where H is the height of $F_{t1\to t2}$, and W is the width of $F_{t1\to t2}$.

3.3 Cross-task consistency

In this section, we connect three independent tasks by the cross-task consistency loss, which includes the rigid consistency loss and the three-view consistency loss.

3.3.1 Rigid consistency loss

We can obtain the rigid flow ($F_{t1\to t2}$) by combining the estimate depth ($D_{t1\to t2}$) and the relative pose ($T_{t1\to t2}$) in Sect. 3.2.1. At the same time, we can get the complete optical flow ($F_{t1\to t2}$) from the optical flow network. Obviously, they should maintain consistency in the rigid region. We formulate the *rigid consistency loss* as

$$\begin{aligned}\mathcal{L}_{\text{ct}}^{\text{rigid}} = & \\ & \frac{1}{|O_{t1\to t2}|} \sum \| F_{t1\to t2} - F_{t1\to t2}^{\text{rigid}} \|_1 R_{t1\to t2} O_{t1\to t2} \\ & + \frac{1}{|1 - O_{t1\to t2}|} \sum \| F_{t1\to t2} - \mathcal{SG}\left(F_{t1\to t2}^{\text{rigid}}\right) \|_1 R_{t1\to t2} \\ & (1 - O_{t1\to t2}).\end{aligned} \tag{9}$$

where $\mathcal{SG}(\cdot)$ stands for stop-gradient, because the rigid flow is more accurate than the predicted optical flow in the occluded area.

3.3.2 Three-view consistency loss

As shown in Fig. 3, we can use the predicted transformation matrix $T_{t2\to t1}$ and $T_{t2\to t3}$ to obtain the trifocal tensor. At the same time, we can use the estimated optical flow $F_{t2\to t1}$ and $F_{t2\to t3}$ for image matching. Therefore, we use the trifocal

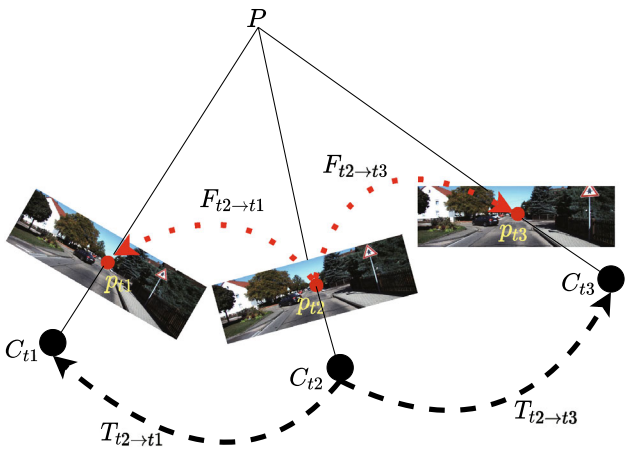


Fig. 3 Schematic illustration of the three-view geometric relationship. We can use transformation matrix $T_{t2\to t1}$ and $T_{t2\to t3}$ to obtain the trifocal tensor, and at the same time we can use optical flow $F_{t2\to t1}$ and $F_{t2\to t3}$ for image matching

tensor incidence relation on point–point–point to introduce a three-view consistency loss.

Let the camera matrices for the three views be $C = [I|\mathbf{0}]$, $C^{'} = [A|\boldsymbol{a_4}]$, and $C^{''} = [B|\boldsymbol{b_4}]$, where A and B are 3×3 matrices, and the vectors a_i and b_i are the i_{th} columns of the respective camera matrices for $i = 1, \ldots, 4$. the $3 \times 3 \times 3$ trifocal tensor could be denoted as $\mathcal{T} = [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]$, where

$$\mathcal{T}_i = \boldsymbol{a}_i \boldsymbol{b}_4^T - \boldsymbol{a}_4 \boldsymbol{b}_i^T. \tag{10}$$

Let the camara matrices of the $t2$ frame be $C_{t2} = K[I|\mathbf{0}]$, we can get

$$\begin{aligned} C_{t1} &= K T_{t2\to t1} = K[\boldsymbol{R}_{t2\to t1}|\boldsymbol{t}_{t2\to t1}], \\ C_{t3} &= K T_{t2\to t3} = K[\boldsymbol{R}_{t2\to t3}|\boldsymbol{t}_{t2\to t3}], \end{aligned} \tag{11}$$

where K denotes the camera intrinsic matrix, $\boldsymbol{R}_{(\cdot)}$ is the rotation component of the predicted transformation matrix $T_{(\cdot)}$, where $\boldsymbol{R} \in \mathbb{SO}(3)$. $\boldsymbol{t}_{(\cdot)}$ is the translation component of the predicted transformation matrix $T_{(\cdot)}$, where $\boldsymbol{t} \in \mathbb{R}^3$. And we can obtain the coordinates $\bar{p}_{t1}$ and $\bar{p}_{t3}$ that match p_{t2} in the $t1$ and $t3$ frames by the estimated optical flows $F_{t2\to t1}(p_{t2})$ and $F_{t2\to t3}(p_{t2})$ as

$$\begin{aligned} \bar{p}_{t1} &= p_{t2} + F_{t2\to t1}(p_{t2}), \\ \bar{p}_{t3} &= p_{t2} + F_{t2\to t3}(p_{t2}). \end{aligned} \tag{12}$$

Then, we normalize the homogeneous coordinates of the matching points, that is,

$$\begin{aligned} \dot{\boldsymbol{p}}_{t1} &= K^{-1}\mathcal{H}(\bar{p}_{t1}), \\ \dot{\boldsymbol{p}}_{t2} &= K^{-1}\mathcal{H}(p_{t2}), \\ \dot{\boldsymbol{p}}_{t3} &= K^{-1}\mathcal{H}(\bar{p}_{t3}), \end{aligned} \tag{13}$$

where $\mathcal{H}(p_{t1})$, $\mathcal{H}(p_{t2})$, $\mathcal{H}(p_{t3})$ are the homogeneous coordinates of p_{t1}, p_{t2}, p_{t3}. Using the trifocal tensor incidence relation on point–point–point, we can get

$$\begin{aligned} &\mathbf{0}_{3\times 3} \backsim [\dot{\boldsymbol{p}}_{t1}]_\times \left(\sum_i \dot{p}_{t2}^i \mathcal{T}_i\right) [\dot{\boldsymbol{p}}_{t3}]_\times, \\ &\dot{\boldsymbol{p}}_{t2} = \left(\dot{p}_{t2}^1, \dot{p}_{t2}^2, \dot{p}_{t2}^3\right)^{\mathrm{T}} \end{aligned} \tag{14}$$

where $[\cdot]_\times$ denoted the cross-product operator, which produces a skew-symmetric matrix from a 3×1 column. From (10), (11), (14), we can get

$$\begin{aligned} \mathbf{0}_{3\times 3} \backsim\ & [\dot{\boldsymbol{p}}_{t1}]_\times \left(\boldsymbol{R}_{t2\to t1} \dot{\boldsymbol{p}}_{t2} \boldsymbol{t}_{t2\to t3}^{\mathrm{T}}\right) [\dot{\boldsymbol{p}}_{t3}]_\times \\ & - [\dot{\boldsymbol{p}}_{t1}]_\times \left(\boldsymbol{t}_{t2\to t1} \dot{\boldsymbol{p}}_{t2} \boldsymbol{R}_{t2\to t3}^{\mathrm{T}}\right) [\dot{\boldsymbol{p}}_{t3}]_\times. \end{aligned} \tag{15}$$

Therefore, we formulate the *three-view consistency loss* as

$$\begin{aligned} \mathcal{L}_{\mathrm{ct}}^{\mathrm{tri}} = \frac{1}{|V_3|} \sum \| & [\dot{\boldsymbol{p}}_{t1}]_\times \left(\boldsymbol{R}_{t2\to t1} \dot{\boldsymbol{p}}_{t2} \boldsymbol{t}_{t2\to t3}^{\mathrm{T}}\right) [\dot{\boldsymbol{p}}_{t3}]_\times \\ & - [\dot{\boldsymbol{p}}_{t1}]_\times (\boldsymbol{t}_{t2\to t1} \dot{\boldsymbol{p}}_{t2} \boldsymbol{R}_{t2\to t3}^{T}) [\dot{\boldsymbol{p}}_{t3}]_\times \|_2 V_3, \end{aligned} \tag{16}$$

$$\begin{aligned} V_3 &= R_{t2\to t1}^h R_{t2\to t3}^h V_{t2\to t1}^{\mathrm{flow}} V_{t2\to t3}^{\mathrm{flow}}, \\ R_{(\cdot)}^h &= [\mathbb{1}](R_{(\cdot)} > \epsilon) \cap [\mathbb{1}](D_{t2} < 0.5\mathcal{M}ax(D_{t2}), \end{aligned} \tag{17}$$

where $[\mathbb{1}]$ is an Iverson bracket, and ϵ is the threshold. $R_{(\cdot)}$ are rigid region weights. $\mathcal{M}ax(D_{t2})$ denote the maximum estimated depth value of the $t2$ frame.

3.4 Forward–backward consistency

For each network, we use forward and backward consistency constraints to further improve each network's performance. Here, the forward–backward consistency loss function includes the *depth forward–backward consistency loss*, the *pose forward–backward consistency loss*, and the *flow forward–backward consistency loss*.

The *depth forward–backward consistency loss* used to constrain the depth discrepancy is formulated as follows:

$$\begin{aligned} &\mathcal{L}_{\mathrm{fb}}^{\mathrm{depth}} = \\ &\frac{1}{|V_{t1\to t2}^{\mathrm{rigid}}|} \sum \left\| \frac{\mathcal{M}(\bar{D}_{t2} V_{t1\to t2}^{\mathrm{rigid}})}{\mathcal{M}(\widehat{D}_{t2} V_{t1\to t2}^{\mathrm{rigid}})} \widehat{D}_{t2} - \bar{D}_{t2} \right\|_1 V_{t1\to t2}^{\mathrm{rigid}} + \\ &\frac{1}{|V_{t1\to t2}^{\mathrm{flow}}|} \sum \left\| \frac{\mathcal{M}(\bar{D}_{t2} V_{t1\to t2}^{\mathrm{flow}})}{\mathcal{M}(\widetilde{D}_{t2} V_{t1\to t2}^{\mathrm{flow}})} \widetilde{D}_{t2} - \bar{D}_{t2} \right\|_1 V_{t1\to t2}^{\mathrm{flow}}, \end{aligned} \tag{18}$$

where $\mathcal{M}(\cdot)$ denote the mean function. $\bar{D}_{t2}$ is the reprojected depth map calculated by D_{t1} and $T_{t1\to t2}$,

$$[\widehat{p}_{t2}, \bar{D}_{t2}(p_{t1})]^T = K T_{t1\to t2} D_{t1}(p_{t1}) K^{-1} \mathcal{H}(p_{t1}). \tag{19}$$

$\widehat{D}_{t2}$ and $\widetilde{D}_{t2}$ are reconstructed depth maps by warping D_{t2} using the synthesized rigid flow $F_{t1\to t2}^{\mathrm{rigid}}$ and the estimated optical flow $F_{t1\to t2}$, respectively. Using the inverse relationship of the transformation matrix $T_{t1\to t2}$ and $T_{t2\to t1}$, we can get the *pose forward–backward consistency loss*, which is formulate as

$$\mathcal{L}_{fb}^{pose} = \sum \left\| T_{t1\to t2}^{-1}, T_{t2\to t1} \right\|_1 + \sum \left\| T_{t2\to t1}^{-1}, T_{t1\to t2} \right\|_1 \tag{20}$$

where $T_{(\cdot)} \in \mathbb{SE}(3)$.

Similarly, we use the optical flow forward–backward consistency check. We formulate the *flow forward–backward consistency loss* as

$$\mathcal{L}_{fb}^{\text{flow}} = \frac{1}{|V_{t1\to t2}^{\text{flow}}|} \sum \| F_{t1\to t2} + \widetilde{F}_{t2\to t1} \|_1 V_{t1\to t2}^{\text{flow}}, \tag{21}$$

where $\widetilde{F}_{t2\to t1}$ is the reconstructed optical flow by warping $F_{t2\to t1}$ using $F_{t1\to t2}$.

3.5 Spatial smoothness priors

In the low texture, homogeneous, and occlusion area of the scene, the photometric loss is insufficient. To address this issue, we introduce edge-aware second smoothness loss weighted by image gradients, which is also used in [24, 44].

We formulate the *depth smoothness loss* as

$$\mathcal{L}_s^{\text{depth}} = \frac{1}{N} \sum_p \sum_{d\in x,y} e^{-\|\nabla_d I(p)\|_1} \|\nabla_d^2 D(p)\|_1, \tag{22}$$

and the *flow smoothness loss* as

$$\mathcal{L}_s^{\text{flow}} = \frac{1}{N} \sum_p \sum_{d\in x,y} e^{-\beta\|\nabla_d I(p)\|_1} \|\nabla_d^2 F(p)\|_1, \tag{23}$$

where β controls the weight of edges, and we set it to 10. N is the total number of pixels. d indexes over partial derivative on x and y directions.

4 Experiments

4.1 Implementation details

Network architecture We adopt similar network designs that align with existing self-supervised learning methods [23, 48, 49]. For the depth network, we use the U-Net architecture. Like the previous works [23, 48–50], the encoder is based on ResNet18 or ResNet50, while the decoder relies on DispNet [51] For the pose network, we use ResNet18/50, modified to accept a pair of color images as input, for feature extraction. Figure 4 shows the schematic diagram of the depth network architecture and the pose network architecture. For the flow network, the structure is based on the PWC-Net [13] that is lightweight and achieves excellent performance in supervised learning of optical flow tasks.

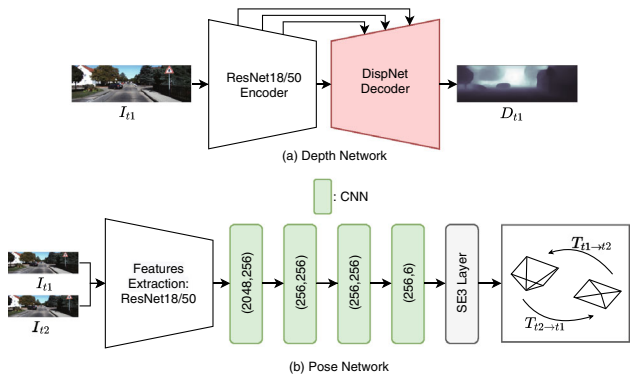


Fig. 4 Network Architecture. **a** Depth Network: the encoder is based on ResNet18 or ResNet50, while the decoder relies on DispNet. **b** Pose Network: we use ResNet18/50, modified to accept a pair of color images as input, for feature extraction

Dataset We use the KITTI dataset [26] in our experiments, a real-world dataset collected from autonomous driving scenes, as the training and evaluation dataset. For the training of depth and flow tasks, we use Eigen et al. [6]'s split of the KITTI raw dataset, which is consistent with related works [20–23, 47, 48, 52]. We evaluate the results of depth estimation using the Eigen et al.'s testing split, consisting of 697 test images. Furthermore, we evaluate the optical flow estimation on the KITTI flow 2015 training set, which contains 200 image pairs. For the odometry task, like Zhou et al. [47], we use sequences 00-08 of the KITTI odometry dataset as the training data and sequences 09, 10 as the testing data.

Training detail Our system is implemented on PyTorch and two NVIDIA Tesla V100 GPUs. We train the networks with a batch size of 8 and an initial learning rate of 10^{-4} using Adam optimization. Hyper-parameters are set to $\beta_1 = 0.95$ and $\beta_2 = 0.999$. We decay the learning rate with a cosine annealing for each epoch. The data are augmented with random scaling, cropping and horizontal flips during training, and the images are resized to 832×256. The encoder part of the depth network and the pose network is initialized with weights trained on ImageNet. The whole training process in our method contains three stages. In the first stage, we only train the optical flow network using $\mathcal{L}_{\text{bc}}^{\text{flow}}$, $\mathcal{L}_{\text{fb}}^{\text{flow}}$, $\mathcal{L}_s^{\text{flow}}$ in an self-supervised manner. In the second stage, we train the depth network and pose networks using $\mathcal{L}_{\text{bc}}^{\text{rigid}}$, $\mathcal{L}_{\text{fb}}^{\text{depth}}$, $\mathcal{L}_{\text{fb}}^{\text{pose}}$, $\mathcal{L}_s^{\text{depth}}$ in an self-supervised manner. In the third stage, we use the three network joint training with the total loss $\mathcal{L}_{total}$ in an self-supervised manner. In all of the three stages, we train the network in 200 epochs with 250 randomly sampled batches in one epoch. The hyper-parameters $\lambda_{\text{bc}}^{\text{rigid}}$, $\lambda_{\text{bc}}^{\text{flow}}$, $\lambda_{\text{ct}}^{\text{rigid}}$, $\lambda_{\text{ct}}^{\text{tri}}$, $\lambda_{\text{fb}}^{\text{depth}}$, $\lambda_{\text{fb}}^{\text{pose}}$, $\lambda_{\text{fb}}^{\text{flow}}$, λ_s^{depth}, λ_s^{flow} are set to be 1.0, 10.0, 0.01, 0.01, 0.5, 0.01, 0.01, 10, 100.

4.2 Experimental results

Optical flow estimation We evaluate the optical flow estimation of our system on the KITTI 2015 stereo/flow training set. As shown in Table 1, we report the average end-point-error (EPE) on non-occluded regions (Noc) and overall regions (All), and Fl score (Fl). The performance of our method is significantly better than other state-of-the-art joint depth pose learning methods. Compared with Wang et al. [24] and Liu et al. [25] that use stereo data for training, our method is also

Table 1 Optical flow estimation results on KITTI 2015 stereo/flow training dataset

Methods	Noc	All	Fl (%)
PWC-Net [13]	–	10.35	–
FlowNet2 [12]	**4.93**	10.06	30.37
RAFT [53]	–	**5.04**	**17.40**
UnFlow-CSS [17]	–	8.10	**23.27**
Back2Future [19]	–	7.04	24.21
EpipolarFlow [18]	**2.98**	**6.02**	–
Geonet [20]	8.05	10.81	–
DF-Net [21]	–	8.98	26.01
EPC++ [54]	–	5.84	–
CC [22]	–	6.21	26.41
GLNet [23]	4.86	8.35	–
Wang et al. [24] (B)	–	5.58	–
Liu et al. [25] (B)	–	5.19	–
Hur et al. [55]	–	7.51	23.49
Wang et al. [56]	–	6.66	23.04
Ours (Flow-only)	4.99	8.71	33.09
Ours (Full)	**3.02**	**4.90**	**16.92**

Top: supervised methods which are trained on synthetic data only. Middle: unsupervised optical flow learning methods. Bottom: joint depth pose learning methods. (B): denotes training with binocular/stereo input pairs. The best performance in each block is highlighted in bold

better. We also report the performance of only training our optical flow network, denoted as "Ours (Flow-only)." After jointing depth pose geometric constraints, "Ours (Full)" improves "Noc" from 4.99 to 3.02, "All" from 8.71 to 4.90 and "Fl" from 33.09 to 16.92%.

The qualitative results of optical flow are shown in Fig. 5. Compared to Liu et al. [25] and "Ours (Flow-only)," "Ours (Full)" performs better on occluded boundary area and the dynamic object area, which benefits from our stronger cross-task constraints.

Monocular depth estimation We report results of depth estimation using the Eigen [6] split of the raw KITTI dataset. The maximum of depth estimation on KITTI split is capped at 80 ms. The results are summarized in Table 2. As shown in Table 2, despite the distance of our method from supervised monocular depth estimation methods, it is clear that our method achieves comparable performance to other self-supervised/unsupervised monocular depth estimation methods.

Camera pose estimation We also evaluate the performance of our method on the KITTI Odometry dataset and compare the results with self-supervised learning methods [22, 47, 48, 52, 58] and geometry-based methods including ORB-SLAM2 [28] (w/ and w/o loop closure). Since monocular systems lack real world scale factor, we align all the poses to the ground-truth with 6-DoF and scale. Except for ORB-SLAM2 (w/ LC), other methods here do not use any loop closure technology. The quantitative and qualitative results of camera pose estimation are shown in Table 3 and Fig. 6. In Table 3, we use the KITTI odometry [39] criteria evaluation that average translational Root-Mean-Square Error (RMSE) drift, t_{err} (%), and average rotational RMSE drift, r_{err} ($^\circ/100\,m$), on length of 100–800 m. Although our method is no better than the geometry-based method ORBSLAM2 [28], we achieve performance improvement over other self-supervised learning systems [22, 47, 48, 52, 58]. In Fig. 6, the recovered trajectories of sequences 9 and 10 are shown. In the XZ-plane, our accumulated drift error is even smaller than ORBSLAM2 (w/ LC), especially on sequence 10 without loopback.

Ablation study The ablation study results of optical flow estimation are shown in Table 4. We can see that joint depth

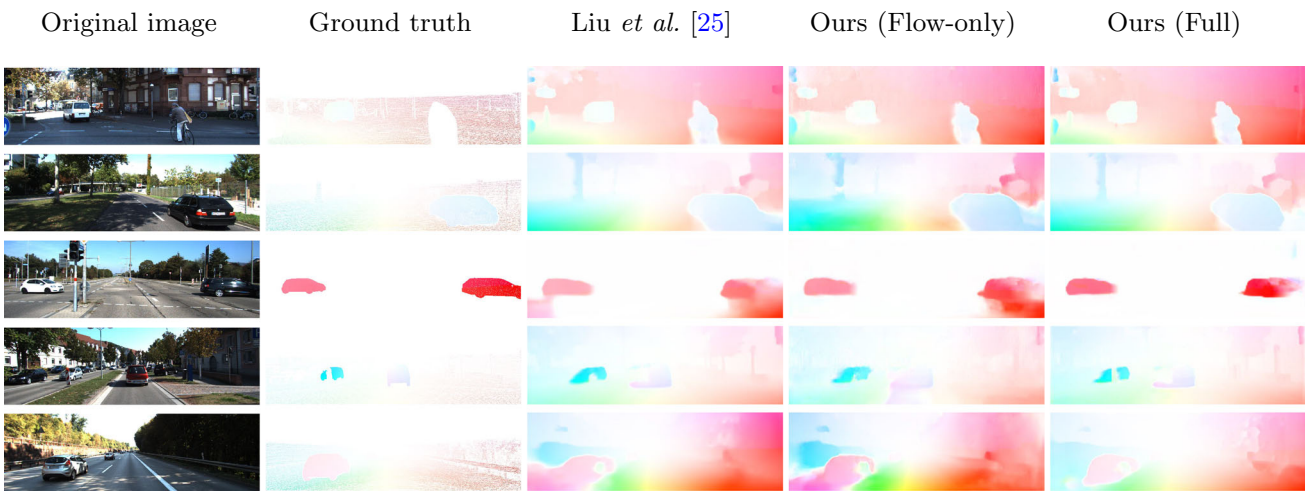


Fig. 5 Qualitative results of optical flow estimation. In the occluded boundary area and the dynamic object area, our method obviously has better performance

Table 2 Monocular depth estimation results on test split of KITTI raw dataset

Methods	Error ↓				Accuracy ↑		
	AbsRel	SqRel	RMS	RMSlog	< 1.25	$< 1.25^2$	$< 1.25^3$
Eigen et al. [6]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [7]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
DepthFormer [36]	**0.052**	0.158	2.143	**0.079**	0.975	**0.997**	**0.999**
BinsFormer [37]	**0.052**	**0.151**	**2.098**	**0.079**	**0.974**	**0.997**	**0.999**
Zhou et al. [47]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Geonet-Resnet [20]	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DF-Net [21]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
CC [22]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
EPC++ [54]	0.141	1.029	5.350	0.216	0.816	0.941	0.976
SC-SfM-Learner [52]	0.137	1.089	5.439	0.217	0.830	0.942	0.975
SC-SfM-Learner (R18) [52]	0.119	0.858	4.949	0.197	0.873	0.957	0.981
GLNet (–ref.) [23]	0.135	1.070	5.230	0.210	0.841	0.948	0.980
Struct2depth (–ref.) [57]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Gordon et al. [49]	0.128	0.959	5.230	0.212	0.845	0.947	0.976
Hur et al. [55]	0.125	0.978	4.877	0.208	0.851	0.950	0.978
Wang et al. [56]	0.140	1.068	5.255	0.217	0.827	0.943	0.977
Monodepth2 [48]	0.115	0.882	4.701	**0.190**	**0.879**	**0.961**	**0.982**
SC-Depth (R18) [50]	0.119	0.857	4.950	0.197	0.863	0.957	0.981
SC-Depth (R50) [50]	**0.114**	**0.813**	**4.706**	0.191	0.873	0.960	**0.982**
Ours (R18)	0.118	0.903	4.809	0.190	0.870	0.959	0.982
Ours (R50)	**0.112**	**0.871**	**4.683**	**0.187**	**0.881**	**0.963**	**0.983**

Top: supervised methods. Middle: self-supervised/unsupervised methods. Bottom: our self-supervised methods. "(R18)" denotes that U-Net extracts features using the ResNet18 encoder and "(R50)" denotes that U-Net extracts features using the ResNet50 encoder. The best performance in each block is highlighted in bold

Table 3 Camera pose estimation results on KITTI odometry dataset

Methods	Seq. 09		Seq. 10	
	t_{err} (%)	r_{err} (°/100m)	t_{err} (%)	r_{err} (°/100 m)
ORB-SLAM2(w/o LC) [28]	9.51	0.24	**2.61**	0.28
ORB-SLAM2(w/ LC) [28]	**2.73**	**0.23**	3.64	**0.26**
Zhou et al. [47]	17.84	6.78	37.91	17.78
Depth-VO-Feat [58]	11.93	3.91	12.45	3.46
CC [22]	7.71	2.32	9.87	4.47
SC-SfM-Learner [52]	11.2	3.35	10.1	4.96
SC-SfM-Learner (R18) [52]	7.31	3.05	7.79	4.90
Monodepth2 [48]	8.61	1.75	9.44	3.14
Ours (R18)	**3.97**	**1.39**	**5.12**	**3.04**

The best performance in each block is highlighted in bold

pose and flow learning, adding the rigid weight/mask, and the three-view consistency loss function can all improve the performance of the overall system. Compared with the situation without the three-view consistency loss, the introduction of the three-view consistency loss has better performance in "Noc" (3.40 vs. 3.02), "All" (5.28 vs 4.90), and "Fl" (18.91% vs. 16.92%).

5 Conclusions

In this paper, we propose a jointly self-supervised learning method of monocular depth estimation, camera pose estimation, and optical flow estimation for 3D geometry

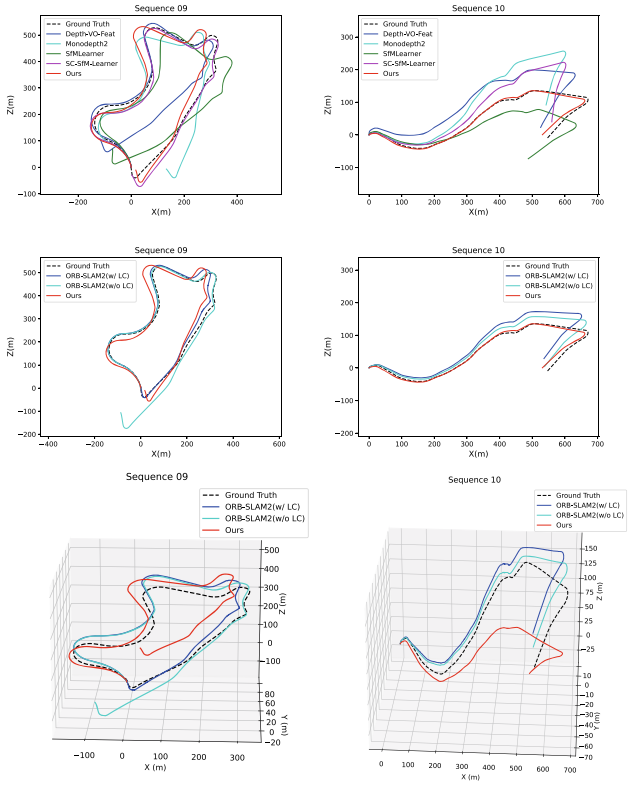


Fig. 6 Qualitative results of pose estimation

Table 4 Ablation study on optical flow estimation

Depth Net	Pose Net	Flow Net	Rigid Weight	$\mathcal{L}_{ct}^{tri}$	Noc	All	Fl (%)
–	–	√	–	–	4.99	8.71	33.09
√	√	√	–	–	3.93	6.21	20.00
√	√	√	√	–	3.40	5.28	18.91
√	√	√	√	√	**3.02**	**4.90**	**16.92**

The best performance in each block is highlighted in bold

understanding with two- and three-view geometric constraints. As we all know, three-view geometric relationships are more informative and robust than two-view geometric relationships. Therefore, in addition to the epipolar geometric constraints used in previous methods, this paper introduces the three-view geometric relations and proposes the three-view consistency loss function, which further improves the consistency of cross-tasks. Experiments show that the three-view consistency loss effectively improves the performance of the system, and our method has better performance in the dynamic object area and the occluded boundary area. Finally, our method has achieved impressive performance in all three subtasks.

Like most other previous methods, our method performs well on the KITTI dataset, but does not work well on realistic data, i.e., it does not have enough generalization capability. Additionally, it does not perform very well in real time.

In future work, we will introduce this system into the current geometry-based SLAM to further improve the overall robustness and practicality of each task. Moreover, we will also extend the proposed self-supervised learning method to engineering applications [59–61].

Acknowledgements This work was supported in part by the STI 2030-Major Projects of China under Grant 2021ZD0201300 and by the National Science Foundation of China under Grant 62276127.

Data availability The datasets generated and analyzed during this study are available in the http://www.cvlibs.net/datasets/kitti/index.php.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

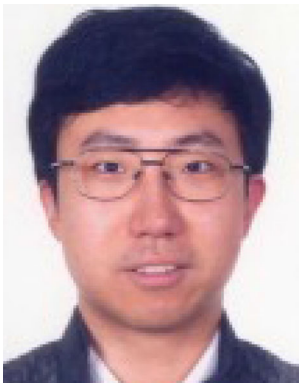
1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
2. Bian, J., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., Cheng, M.-M.: Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: IEEE conference on computer vision and pattern recognition (CVPR), pp. 4181–4190 (2017)
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
4. Horn, B.K., Schunck, B.G.: Determining optical flow. In: Techniques and Applications of Image Understanding, vol. 281, pp. 319–331 (1981). International Society for Optics and Photonics, USA
5. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2439 (2010). IEEE
6. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Neural Information Processing Systems (NeurIPS) (2014)
7. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. IEEE Trans. Pattern Recognit. Mach. Intell. (PAMI) (2016). https://doi.org/10.1109/TPAMI.2015.2505283
8. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
9. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2043–2050 (2017). IEEE
10. Wang, S., Clark, R., Wen, H., Trigoni, N.: End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. Int. J. Robot. Res. (IJRR) **37**(4–5), 513–542 (2018)
11. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015)
12. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
13. Sun, D., Yang, X., Liu, M.-Y., Kautz, J.: Pwc-net: CNNS for optical flow using pyramid, warping, and cost volume. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

14. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4161–4170 (2017)
15. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision (ECCV) (2016). Springer
16. Godard, C., Aodha, O.M., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
17. Meister, S., Hur, J., Roth, S.: UnFlow: unsupervised learning of optical flow with a bidirectional census loss. In: Association for the Advancement of Artificial Intelligence (AAAI), New Orleans, Louisiana (2018)
18. Zhong, Y., Ji, P., Wang, J., Dai, Y., Li, H.: Unsupervised deep epipolar flow for stationary or dynamic scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12095–12104 (2019)
19. Janai, J., Guney, F., Ranjan, A., Black, M., Geiger, A.: Unsupervised learning of multi-frame optical flow with occlusions. In: European Conference on Computer Vision (ECCV), pp. 690–706 (2018)
20. Yin, Z., Shi, J.: GeoNet: unsupervised learning of dense depth, optical flow and camera pose. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
21. Zou, Y., Luo, Z., Huang, J.-B.: DF-Net: unsupervised joint learning of depth and flow using cross-task consistency. In: European Conference on Computer Vision (ECCV) (2018)
22. Ranjan, A., Jampani, V., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive Collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: IEEE International Conference on Computer Vision (ICCV), pp. 7063–7072 (2019)
24. Wang, Y., Yang, Z., Wang, P., Yang, Y., Luo, C., Xu, W.: Joint unsupervised learning of optical flow and depth by watching stereo videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
25. Liu, L., Zhai, G., Ye, W., Liu, Y.: Unsupervised learning of scene flow estimation fusing with local rigidity. In: Association for the Advancement of Artificial Intelligence (AAAI), pp. 876–882 (2019)
26. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. (IJRR) **32**(11), 1231–1237 (2013)
27. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3D reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium (IV), pp. 963–968 (2011). IEEE
28. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. (TRO) **33**(5), 1255–1262 (2017)
29. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular slam. In: European Conference on Computer Vision (ECCV), pp. 834–849 (2014). Springer
30. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Recognit. Mach. Intell. (PAMI) **40**(3), 611–625 (2017)
31. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV), pp. 239–248 (2016). IEEE
32. Kendall, A., Grimes, M., Cipolla, R.: Posenet: a convolutional network for real-time 6-dof camera relocalization. In: IEEE International Conference on Computer Vision (ICCV), pp. 2938–2946 (2015)
33. Li, R., Liu, Q., Gui, J., Gu, D., Hu, H.: Indoor relocalization in challenging environments with dual-stream convolutional neural networks. IEEE Trans. Autom. Sci. Eng. **15**(2), 651–662 (2017)
34. Costante, G., Mancini, M., Valigi, P., Ciarfuglia, T.A.: Exploring representation learning with CNNS for frame-to-frame ego-motion estimation. IEEE Robot. Autom. Lett. **1**(1), 18–25 (2015)
35. Clark, R., Wang, S., Wen, H., Markham, A., Trigoni, N.: Vinet: visual-inertial odometry as a sequence-to-sequence learning problem. In: Association for the Advancement of Artificial Intelligence (AAAI) (2017)
36. Li, Z., Chen, Z., Liu, X., Jiang, J.: Depthformer: exploiting long-range correlation and local information for accurate monocular depth estimation. arXiv preprint arXiv:2203.14211 (2022)
37. Li, Z., Wang, X., Liu, X., Jiang, J.: Binsformer: revisiting adaptive bins for monocular depth estimation. arXiv preprint arXiv:2204.00987 (2022)
38. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
39. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361 (2012). IEEE
40. Butler, D., Wulff, J., Stanley, G., Black, M.: MPI-sintel optical flow benchmark: Supplemental material. In: MPI-IS-TR-006, MPI for Intelligent Systems (2012). Citeseer
41. Pilzer, A., Lathuiliere, S., Sebe, N., Ricci, E.: Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9768–9777 (2019)
42. Ahmadi, A., Patras, I.: Unsupervised convolutional neural networks for motion estimation. In: IEEE International Conference on Image Processing (ICIP), pp. 1629–1633 (2016). IEEE
43. Jason, J.Y., Harley, A.W., Derpanis, K.G.: Back to basics: unsupervised learning of optical flow via brightness constancy and motion smoothness. In: European Conference on Computer Vision (ECCV), pp. 3–10 (2016). Springer
44. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4884–4893 (2018)
45. Godard, C., Mac Aodha, O., Brostow, G.: Digging into self-supervised monocular depth estimation. arXiv preprint arXiv:1806.01260 (2018)
46. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
47. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
48. Godard, C., Aodha, O.M., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: IEEE International Conference on Computer Vision (ICCV) (2019)
49. Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: IEEE International Conference on Computer Vision (ICCV), pp. 8977–8986 (2019)
50. Bian, J.-W., Zhan, H., Wang, N., Li, Z., Zhang, L., Shen, C., Cheng, M.-M., Reid, I.: Unsupervised scale-consistent depth learning from video. Int. J. Comput. Vis. **129**(9), 2548–2564 (2021)
51. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4040–4048 (2016)

52. Bian, J.-W., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.-M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: Neural Information Processing Systems (NeurIPS) (2019)
53. Teed, Z., Deng, J.: Raft: recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision (ECCV), pp. 402–419 (2020). Springer
54. Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., Yuille, A.: Every pixel counts ++: joint learning of geometry and motion with 3d holistic understanding. IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI) **PP**(99), 1–1
55. Hur, J., Roth, S.: Self-supervised monocular scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7396–7405 (2020)
56. Wang, G., Zhang, C., Wang, H., Wang, J., Wang, Y., Wang, X.: Unsupervised learning of depth, optical flow and pose with occlusion from 3D geometry. In: IEEE Transactions on Intelligent Transportation Systems (TITS) (2020)
57. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: leveraging structure for unsupervised learning from monocular videos. In: Association for the Advancement of Artificial Intelligence (AAAI), vol. 33, pp. 8001–8008 (2019)
58. Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
59. Liang, Y., He, F., Zeng, X., Luo, J.: An improved loop subdivision to coordinate the smoothness and the number of faces via multi-objective optimization. Integrated Computer-Aided Engineering (Preprint), pp. 1–19 (2022)
60. Wu, Y., He, F., Zhang, D., Li, X.: Service-oriented feature-based data exchange for cloud-based design and manufacturing. IEEE Trans. Serv. Comput. **11**(2), 341–353 (2015)
61. Song, Y., He, F., Duan, Y., Liang, Y., Yan, X.: A kernel correlation-based approach to adaptively acquire local features for learning 3D point clouds. Comput.-Aided Des. **146**, 103196 (2022)

Xiaoliang Liu obtained his B.S. degree in communication engineering from Fujian University of Technology, Fujian, China. Currently, he is pursuing his Ph.D. in the Department of Computer Science and Technology at Nanjing University, Jiangsu, China. His research focuses on artificial neural networks, computer vision, and robot intelligence.

Furao Shen a member of IEEE, received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from theTokyo Institute of Technology, Tokyo, Japan, in 2006. He is currently a full professor of School of Artificial Intelligence with Nanjing University. His current research interests include neural computing and robotic intelligence.



Jian Zhao received the B.S. degree from Nanjing University, Nanjing, China, the M.Sc. degree from Hamburg University of Technology, Hamburg, Germany, and the Dr. Sc. degree in electrical engineering from Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. From 2010 to 2015, he was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. Currently, he is an associate professor with the School of Electronic Science and Engineering in Nanjing University. His research interests include deep neural networks, mathematical optimization, and wireless communication networks.

Changhai Nie is a professor of software engineering with the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University. His research interests include software testing and search-based software engineering, especially in combinatorial testing, search-based software testing, software testing methods comparison and combination, etc.