Un-VDNet: unsupervised network for visual odometry and depth estimation

Xuyang Meng[a], **Chunxiao Fan**[a], **Yue Ming**[a,*], **Yuan Shen**[b], **Hui Yu**[c]
[a]Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Haidian District, Beijing, China, 100876
[b]Autonomous Driving Center, Tencent Technology (Beijing) Co., Ltd. Beijing, China, 100080
[c]Portsmouth University, Portsmouth, UK, PO1 2UP

Abstract. Monocular visual odometry and depth estimation plays an important role in augmented reality and robots applications. Recently, deep learning technologies have been widely used in these areas. However, most existing works utilize supervised learning which requires large amounts of labeled data, and assumes that the scene is static. In this paper, we propose a novel framework, called as Un-VDNet, based on unsupervised convolutional neural networks (CNNs) to predict camera ego-motion and depth maps from image sequences. The framework includes three sub-networks (PoseNet, DepthNet, and FlowNet), and learns temporal motion and spatial association information in an end-to-end network. Specially, we propose a novel pose consistency loss to penalize errors about the translation and rotation drifts of the pose estimated from the PoseNet. Furthermore, a novel geometric consistency loss, between the structure flow and scene flow learned from the FlowNet, is proposed to deal with dynamic objects in the real-world scene, which is combined with spatial and temporal photometric consistency constraints. Extensive experiments on the KITTI and TUM datasets demonstrate that our proposed Un-VDNet outperforms the state-of-the-art methods for visual odometry and depth estimation in dealing with dynamic objects of outdoor and indoor scenes.

Keywords: visual odometry, depth estimation, unsupervised CNNs, consistency constraint.

*Yue Ming, myname35875235@126.com

1 Introduction

Visual odometry and depth estimation is a key enabling technique of robots and autonomous vehicles, which effectively captures environment information for location, navigation, and target tracking, *etc.*[1] and has attracted more and more attention recently.[2] Although recent researches in visual odometry and depth estimation have achieved remarkable progress, there are still many challenges in accuracy and robustness due to dynamic scenes, *etc.*

Classical Structure-from-Motion (SfM) methods have been researched on visual odometry and depth estimation for several decades,[3,4] which achieve more effective performance in reconstruction and navigation systems,[5] and capture the semantic information of the scene.[6] However, traditional SfM methods, extracting the low-level features, cannot deal with outliers and mismatches in texture-less regions,[7] and motion drifts resulted by dynamic objects. In order to overcome these limitations of classical SfM, many end-to-end model learning methods have been proposed to estimate the camera trajectories and depth maps simultaneously through learning convolutional neural network (CNN) features benefited from big-data.[8] Methods based on CNNs have made large progress in pixel-level estimation tasks,[9] so that they are more robust in camera motion and challenging environments.[10] However, most deep learning methods require large amounts of labeled data for supervision or assume a static and general environment. In fact, there are usually many dynamic objects in the real-world, so they are unsuitable when dynamic objects account for a large proportion in the scene.[11]

In this paper, we propose a novel unsupervised deep learning framework, Un-VDNet, for simultaneously estimating visual odometry and depth from monocular image sequences, which effectively deals with dynamic objects in both outdoor and indoor scenes. The Un-VDNet consists of three sub-networks, PoseNet, DepthNet, and FlowNet. The PoseNet is constructed to learn camera ego-motion between adjacent frames and estimate camera trajectories for visual odometry. It is constrained by temporal photometry and trained with a forward-backward consistency and a pose consistency loss. The DepthNet is proposed to estimate pixel-level depth maps with spatial photometric consistency constraints. We formulate structure-flow based on camera poses and depth maps predicted from the PoseNet and DepthNet, respectively. The geometric estimator of our framework, FlowNet, predicts the scene optical flow to assist in handle dynamic objects. We propose a novel geometric consistency loss to train our network between structure-flow and scene optical flow, which effectively improves the performance of our model and enhances our predictions on camera poses and depth maps. The Un-VDNet is trained on stereo image sequences and tested on monocular sequences. The illustration of our proposed framework is shown in Figure 1.

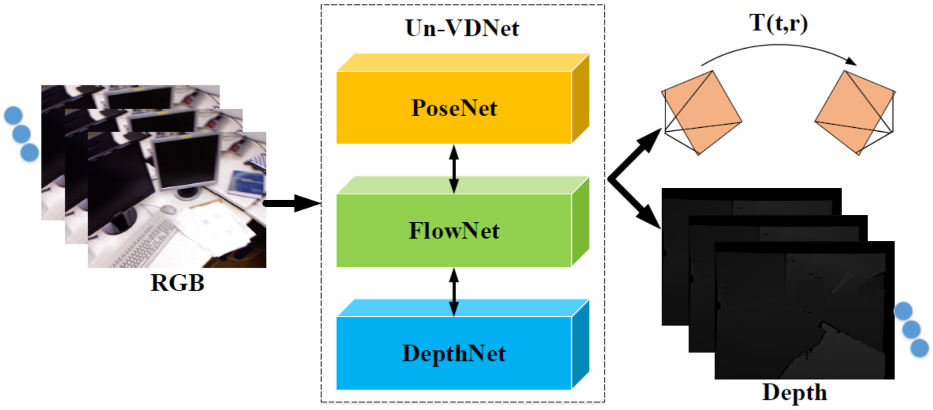


Fig 1 Illustration of our proposed framework Un-VDNet. The network consists of three sub-networks, PoseNet. DepthNet, and FlowNet. It is trained on unlabeled RGB images in unsupervised manner and estimates the camera pose T(t,r) and image depth, where T $\in \mathbb{R}^{4\times4}$ is transform matrix, t $\in \mathbb{R}^3$ is translation, and r $\in \mathbb{R}^3$ is rotation.

Our contributions can be summarized as follows :

1) we propose a novel unsupervised network, Un-VDNet, to estimate camera ego-motion and depth maps, which not only predicts the static structure without any label but also handles the problems caused by dynamic objects in the outdoor and indoor scenes for the first time;

2) we construct a novel loss based on pose consistency to enhance the camera translation and rotation estimations;

3) we construct a geometric estimator trained with a novel geometric consistency loss between scene optical flow and structure-flow to overcome challenges on motion-blur in dynamic scenes.

Experiments on KITTI dataset[12] and TUM dataset[13] demonstrate the effectiveness of our proposed framework in both outdoor and indoor dynamic scenes, and our unsupervised network outperforms previous CNNs methods in visual odometry and depth estimation from photometry to geometry.

The rest of this paper is organized as follows. Section 2 reviews related works about visual odometry and depth estimation based on CNNs. Section 3 introduces our proposed framework and loss functions. Section 4 presents our experiments and results on KITTI dataset[12] and TUM dataset[13] about visual odometry and depth estimation. Our conclusions are summarized in Section 5.

2 Related Work

Visual odometry has been widely researched after first proposed by Nister[14] for ego-motion and depth estimation. Traditional methods not only need to capture and match hand-craft features, but also have a heavy cost.[15] In order to learn camera pose and depth map more efficiently, methods based on CNNs have been introduced recently.[16] In this section, we summarize some related works about visual odometry and depth estimation based on CNNs.

2.1 Visual Odometry Based on CNNs

Visual odometry is the process of predicting camera ego-motion by analyzing the multi-view geometry between adjacent frames. Compared with traditional visual odometry based on features or direct methods, approaches based on deep learning estimated the camera pose in an end-to-end model and didn't rely on complex geometric operations, which were more intuitive and concise. Kendall $et\ al.$[17] were the first to propose an end-to-end CNN, pose-net, to regress the 6-DOF camera pose for re-localization from a single RGB image without additional engineering or graph optimization. The pose-net was faster than traditional ego-motion estimation methods based on SfM, but its accuracy on translation and orientation estimation was lower than mainstreams. Costante $et\ al.$[18] applied CNNs to learn both the optimal features and the best estimator to estimate the F2F camera motion with dense optical flow. Their method outperformed pose-net[17] in translation estimation but had poor performance in rotation. Handa $et\ al.$ [19] designed an end-to-end network by extending a spatial transform network to regress the classical computer vision methods. They constructed geometric vision with a neural network (gvnn) including a global transformation and pixel transformation kernel M estimator to estimate the camera pose based on RGB-D data. Their approach ensured that the loss functions are as close as 0 when converging, and the missing pixels could be properly processed. Wu *et al.*[20] imported BranchNet including two branches for orientation and translation prediction respectively based on pose-net,[17] and they proposed the Euler6 to represent orientation. But the method was limited in some scenarios where depth is unavailable.

In addition to above-mentioned supervised CNNs, there were also many unsupervised methods for visual odometry. SfM-Learner,[21] a multi-view end-to-end network proposed by Google, consisted of single-view depth and multi-view pose networks with view synthesis as the supervisory signal. However, it caused a larger error when dynamic objects appeared in the scene. Li *et al.*[22] proposed a monocular visual odometry pipeline UnDeepVO, which applied two salient features: an unsupervised deep learning scheme, and an absolute scale recovery. They trained the network to recover the scale using stereo images based on spatial and temporal dense information and tested on consecutive monocular images. Both their mean translational error is three times and the mean rotational error is one time lower than SfM-Learner,[21] but they also ignored dynamic objects in the scene. Therefore, we propose an unsupervised network with a pose consistency loss to reduce errors of translation and rotation estimation, and a geometry consistency loss to deal with dynamic scenes.

2.2 Depth Estimation Based on CNNs

Accurate depth information plays an important role in 3D reconstruction and SLAM. Classic SfM methods usually take stereo matching algorithms and the triangulation principle to calculate the disparity and estimate the view depth.[23] However, its scope and accuracy are limited in dynamic scenes. Depth estimation based on CNNs is popular with researchers along with the development of deep learning, and it can overcome the challenges of SfM.

Eigen $et\ al.$[12] were the first to propose a multi-scale network to predict depth maps, but most maps contained outliers, especially in objects edges, windows, and reflective surfaces. Tateno $et\ al.$ [7] put forward a propagation system, CNN-SLAM, extending ResNet-50[24] to a full convolution network. They minimized the soft-max layer and entropy loss functions using back-propagation and stochastic gradient descent (SGD). The pipeline combined the depth predicted by CNN and SLAM respectively to estimate depth map. Garg $et\ al.$[25] performed an unsupervised encoder-decoder network to minimize the color constancy error for well-warps. The reconstruction loss for their encoder is the photometric error. However, they assumed that the scene was static and ignored dynamic objects. Sudheendra $et\ al.$[26] proposed a geometry-aware network, SfM-Net, which converted depth prediction into a dense flow field in videos. The network was trained with the re-projection photometric error and depth provided by RGB-D sensors. Although the log RMSE was 0.31 with respect to ground truth, it had low tolerance to dynamic regions. Godard $et\ al.$ [27] carried out a CNN network named monodepth using parasitic geometric constraints to generate disparity map. Their model learned to perform a single image depth estimation with good performance and robustness. DeMoN[10] was the first CNNs to predict single view depth from two unconstrained images, whose core part was an iterative encoder-decoder network. It was trained on spatial relative differences to estimate depth maps. But its flexibility was not as good as the traditional SfM methods when the camera's internal parameters changed. Zhan $et\ al.$[28] proposed an unsupervised single view depth estimation network, Depth-VO-Feat, trained with a deep feature reconstruction loss with self-embedded depth features. However, they also assumed that the scene was static and there were no dynamic objects in the view.

How to effectively treat dynamic objects in the scene is still a challenge task for visual odometry and depth estimation.

3 Our Proposed Method

In this section, we firstly elaborate an overview of the proposed framework, Un-VDNet, for simultaneously predicting camera ego-motion and depth maps from a monocular sequence. Then we illustrate details about its three sub-networks: PoseNet, DepthNet, and FlowNet, and training losses constrained by photometric and geometric consistency.

3.1 Overview of Un-VDNet

The proposed framework, Un-VDNet, as shown in Figure 2, perceives the 3D scene structure in an unsupervised learning. Un-VDNet is trained on stereo sequences and tested on a monocular sequence so it takes advantage of the temporal and spatial photometric consistency. Our network is divided into three modules: PoseNet, DepthNet, and FlowNet, to perform visual odometry and depth estimation in both outdoor and indoor scenes. Camera pose **T (t,r)** and depth d are regressed from PoseNet and DepthNet respectively with photometric consistency constraints, and then fused to generate the structure flow f^s. The FlowNet predicts scene optical flow f^f, which is minimized to the structure flow f^s with the geometric consistency. Since each sub-network constructs a special task, 3D scene understanding becomes much easier. These three parts are trained in an unsupervised manner jointly and regressed simultaneously. We utilize view reconstruction as the fundamental supervision for our network from photometry and geometry without any other labels.

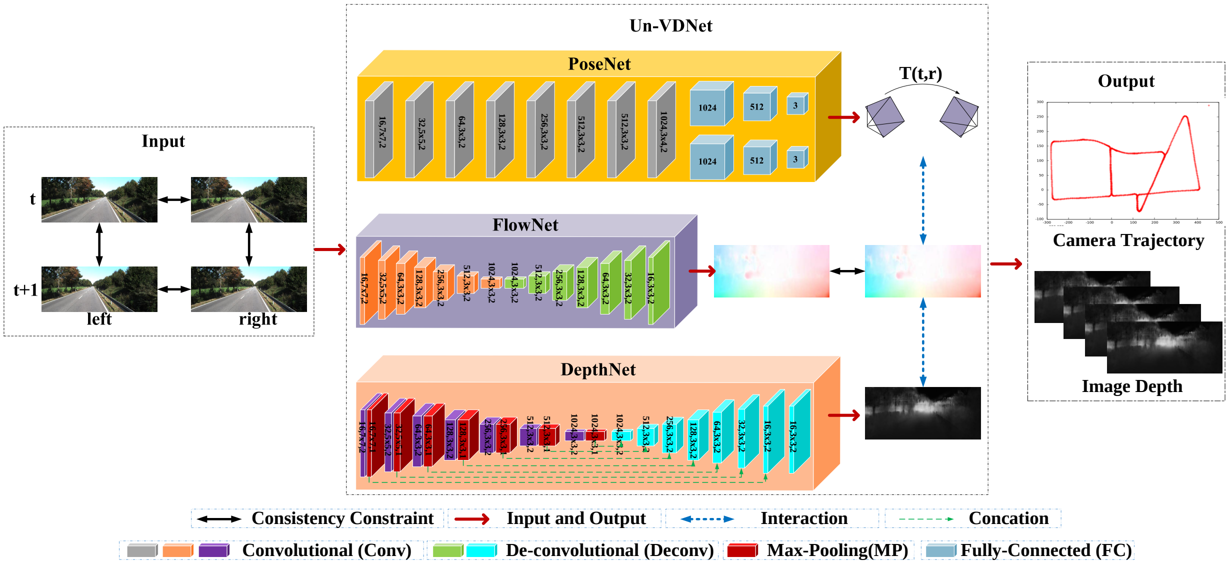


Fig 2 Architecture of our proposed framework, Un-VDNet. The framework consists of three sub-networks: a PoseNet for camera pose, a DepthNet for depth map, and a FlowNet for flow estimation. Details are described in Section 3.

3.2 PoseNet

The PoseNet is based on ResNet-50[24] architecture with two groups of fully-connected (FC) layers. It takes monocular sequences as input and estimates the 6-DOF camera pose **T (t,r)** between adjacent frames, where **t** is 3D translation vector and **r** is 3D Euler angle.[29] To better predict translation and rotation in an unsupervised learning, we add two groups of fully-connected layers behind the convolutional layers.

Forward – Backward Consistency Loss. The PoseNet is constrained by temporal photometric consistency between two consequent monocular images. To learn camera ego-motion, we minimize the forward-backward photometric consistency loss between t th and (t+1) th frame. We denote p_t as one pixel in I^t, p_{t+1} as the corresponding pixel in I^{t+1}, and t th and (t+1) th pair-wise images as $\{I_p^t, I_p^{t+1}\}$, taken by calibrated monocular with camera intrinsics **K**. The relationship between p_t and p_{t+1} can be formulated as:

$$p_{t+1} = \mathbf{K}\mathbf{T}_{t,t+1} d\left(p_t\right) \mathbf{K}^{-1} p_t \tag{1}$$

where $\mathbf{T}_{t,t+1}$ is the camera 6-DOF pose from the frame I_p^t to frame I_p^{t+1}, $d\left(p_t\right)$ is depth value of the pixel p_t in the frame I_p^t. $\mathbf{T}_{t,t+1}$ is a transformation matrix computed from 3D translation vector $\mathbf{t}$ and 3D rotation vector $\mathbf{r}$.

We can synthesize $I_p^{t'}$ from I_p^{t+1}, and $I_p^{t+1'}$ from I_p^t with predicted camera pose $\mathbf{T}_{t,t+1}$ and image depth $d\left(p_t\right)$ through $Spatial\ Transformer$,[30] which is similar to Figure 3. For simplicity, we define synthesized images $I_p^{t'}$ and $I_p^{t+1'}$ as I_p^s, original images I_p^t and I_p^{t+1} as I_p^o. Therefore, we minimize the forward-backward consistency loss fused by a $\mathbf{L}_1$ norm and a **SSIM** term between original images I_p^o and synthesized images I_p^s as follows:[31]

$$\mathbf{L}_p = \sum_p \alpha \frac{1 - \textbf{SSIM}\left(I_p^o, I_p^s\right)}{2} + \sum_p \left(1-\alpha\right) ||I_p^o - I_p^s||_1 \tag{2}$$

where $\left\{I_p^o, I_p^s\right\}$ are original and synthesized pair-wise images, I_p^s is the synthesized image from original image by $Spatial\ Transformer$, and α is a weight between $\mathbf{L}_1$ norm and **SSIM** term.

In most regression problems, the cost function is generally $\mathbf{L}_1$ norm for the Manhattan distance or $\mathbf{L}_2$ norm for the Euclidean distance. $\mathbf{L}_1$ is sensitive to small errors and calculates all pixels in patches. $\mathbf{L}_2$ can suppress large errors, but it is highly tolerant of small errors. The Structural Similarity Index Metric (SSIM) proposed by Wang $et\ al.$[32] is sensitive to local information changes and can be derived, but it only calculates the central pixel of the patch and then applies it to each pixel in the patch. The larger the SSIM, the better the visual effect of the image. To be closer to the perception of the human vision system and estimate the structural similarity of images,[33] we combine the $\mathbf{L}_1$ norm and **SSIM** term like literature.[31] **SSIM** term for pixel p is defined as :[32]

$$\mathbf{SSIM}\,(p) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{3}$$

where μ_x and μ_y are means, and σ_x^2, σ_y^2, and σ_{xy} are deviations, which are computed with a Gaussian filter with standard deviation. C_1 and C_2 are two different constants.

Pose Consistency Loss. Apart from the forward-backward consistency loss for I_p^t and I_p^{t+1} images, we propose a novel pose consistency loss between the predicted transformations $\mathbf{T_o}(\mathbf{t}, \mathbf{r})$ of original image and $\mathbf{T_s}(\mathbf{t}, \mathbf{r})$ of synthesized image. These estimated camera poses from the original and the synthesized image should be equivalent ideally. That is, the predicted transformations $\mathbf{T_o}(\mathbf{t}, \mathbf{r})$ and $\mathbf{T_s}(\mathbf{t}, \mathbf{r})$ are as close as possible. In order to penalize the difference between them and improve the adaptive capabilities of the PoseNet, we formulate a novel cost function about 3D translation and rotation vectors additionally:

$$\mathbf{L}_{tr} = \theta||\mathbf{t_o} - \mathbf{t_s}||_\mathbf{2}^\mathbf{2} + (\mathbf{1} - \theta)||\mathbf{r_o} - \mathbf{r_s}||_\mathbf{2}^\mathbf{2} \tag{4}$$

where θ is the weight of translation and rotation consistency, $(\mathbf{t_o}, \mathbf{r_o})$ is the estimated pose of the original image, and $(\mathbf{t_s}, \mathbf{r_s})$ is the estimated pose of the synthesized image obtained from $Spatial\ Transformer$.[30] These 3D vectors $\mathbf{t_o}$ and $\mathbf{t_s} \in \mathbb{R}^\mathbf{3}$ represent translations, and $\mathbf{r_o}$ and $\mathbf{r_s} \in \mathfrak{so}^\mathbf{3}$ are axis-angle representations.

In short, we mainly train the PoseNet by a forward-backward consistency loss and a pose consistency loss to learn visual odometry. However, it should be pointed out that we assume the scene is static when we utilize a photometric consistency loss. Dynamic objects in the scene may cause large drifts during the translation and rotation learning. Therefore, we construct a geometric estimator, FlowNet, to deal with dynamic scenes and enhance the PoseNet, which is described in Section 3.4.

3.3 DepthNet

For single-view depth estimation, we adopt the network similar as DispNet[34] with multi-scale predictions based on the encoder-decoder architecture. It takes stereo sequences as training data and estimates depth maps between left and right images. At test, our DepthNet predicts the disparity D(p) for a monocular image, and then converts to a depth map $d(p)$ with the known camera intrinsics.

$\mathbf{Left-Right\ Consistency\ Loss}$. The DepthNet is constrained by spatial photometric consistency between stereo sequences to predict single-view depth. We minimize the left-right consistency loss between left and right frames. We denote stereo pair-wise images as $\left\{I_p^l, I_p^r\right\}$, taken by calibrated stereo cameras with focal length f and baseline B. Supposing the inverse depth at pixel p is $z\left(p\right)$ predicted from our network, the view disparity $D\left(p\right)$ equals to $fB/z\left(p\right)$. We apply "$Spatial\ Transformer$"[30] to synthesize another image from one image with disparity $D\left(p\right)$ as shown in Figure 3, so that we obtain the synthesized pair-wise images $\left\{I_p^{l'}, I_p^{r'}\right\}$. Like the PoseNet, we denote original images as I_p^o and synthesized images as I_p^s. Therefore, we introduce the left-right photometric consistency loss $\mathbf{L_d}$ which is similar to Eq. 2 .

$\mathbf{Disparity\ Smoothness\ Loss}$. There are depth discontinuities in image gradients due to some sudden changes in image gray-scale, and multiple disparities can produce the same warps. We thus need prior knowledge to obtain a unique depth map. Following the method adopted by

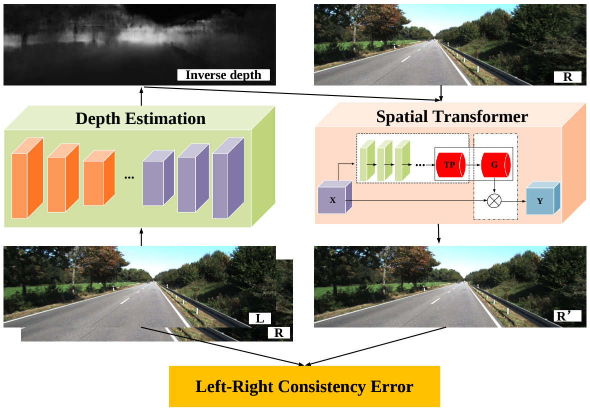


Fig 3 The process of left image warping. The DepthNet predicts inverse depth from left image L. Then we use Spatial Transformer (STN) [30] to synthesize a new right image R' with inverse depth map and raw right image R. In STN, X and Y are feature maps, TP is transformation parameters, and G is sampling grid. We can synthesize a new left image L' in the same way.

Zhou $et\ al.$[21] and Zhan $et\ al.$,[28] we formulate this cost through a disparity smoothness loss with edge-aware terms,

$$\mathbf{L}_{smooth} = \sum_{p} \left(|\partial_x D_p| \cdot e^{-||\partial_x I_p||} + |\partial_y D_p| \cdot e^{-||\partial_y I_p||} \right) \tag{5}$$

where ∂_x and ∂_y are gradients in horizontal and vertical direction respectively, and D_p is disparity map.

In short, the DepthNet is trained with a photometric consistency and a disparity smooth loss to learn smooth depth maps. Similar to the PoseNet, the left-right consistency constraint ignores dynamic objects such as cars and pedestrians in the real-world scenes, which are disarrangement and lead to motion-blur. Therefore, the FlowNet with geometric estimation efficiently strengthen the PoseNet and the DepthNet in dynamic scenes.

3.4 FlowNet

Since the PoseNet and the DepthNet depend on the assumption that the scene is static and ignore dynamic objects, these regressions about camera pose and depth map have large errors with respect to the ground truth. The universal optical flow is commonly applied to off-the-shelf deep learning networks,[35] which can directly simulate unconstrained motion. The pose and depth estimation give satisfactory results in most static scene, while the flow estimation focuses on localizing dynamic scenes, such as cars and persons. Therefore, we propose a geometric estimator, FlowNet, to deal with dynamic objects in the view to enhance visual odometry and depth estimation. The FlowNet of our Un-VDNet is adopted from the ResFlowNet[36] which estimates the scene optical flow $f_t^f\left(p\right)$. Based on our estimated the 6-DOF camera pose $\mathbf{T}_{t,t+1}$ between t th and $(t+1)$ th images and depth map $d_t\left(p\right)$, we obtain the structure flow $f_t^s\left(p\right)$ through

$$f_t^s\left(p\right) = \mathbf{K}\mathbf{T}_{t,t+1} d_t\left(p\right) \mathbf{K}^{-1} p_t - p_t \tag{6}$$

where $\mathbf{K}$ is calibrated camera intrinsics, $\mathbf{T}_{t,t+1}$ is the camera 6-DOF transformation from the frame I_t to frame I_{t+1}, $d_t\left(p\right)$ is depth map, and p_t is the homogeneous coordinates of a pixel in frame I_t. Thus, when we obtain the scene optical flow $f_t^f\left(p\right)$ and the estimated camera pose $\mathbf{T}_{t,t+1}$, we can update the t th depth map $d_t\left(p\right)$ based on Eq.(6). Similarly, we can also adjust the transformation $\mathbf{T}_{t,t+1}$ between t th and $(t+1)$ th images with the scene optical flow $f_t^f\left(p\right)$ and the estimated depth map $d_t\left(p\right)$.

Geometry Consistency Loss. In order to better learn in dynamic scenes, we constrain the structure flow $f_t^s\left(p\right)$ and scene optical flow $f_t^f\left(p\right)$ from the FlowNet with geometric flow consistency. We formulate a geometric consistency loss between them, which is constructed by $\mathbf{L}_1$

norm as follow,

$$\mathbf{L}_f = \sum_p \left(||f_t^f\left(p\right) - f_t^s\left(p\right)||_1 \right) \tag{7}$$

It can not only rectify wrong prediction in the PoseNet and the DepthNet but also refine imperfect results which are caused by dynamic objects in the scene.

The final loss function of our proposed framework contains previous losses together as follows:

$$\mathbf{L}_{final} = \sum_p \left(\lambda_p \left(\mathbf{L}_p + \mathbf{L}_{tr} \right) + \lambda_d \left(\mathbf{L}_d + \mathbf{L}_{smooth} \right) + \lambda_f \mathbf{L}_f \right) \tag{8}$$

where λ_p, λ_d, and λ_f are the loss weights for each term.

3.5 Network Architecture

Our Un-VDNet mainly contains three sub-networks, the PoseNet, the DepthNet, and the FlowNet. Since both DepthNet and FlowNet construct pixel-level predictions, we adopt the encoder-decoder network architecture as a backbone. The encoder follows the basic structure of ResNet-50[24] due to its feed-forward connection manners and effectiveness for pixel-level learning tasks as follow previous works. The decoder is made up of deconvolution layers to enlarge the spatial feature maps to full scale as input. To preserve both global high-level and local detailed information, we use skip connections between encoder and decoder parts at different corresponding resolutions. Both the depth map and scene optical flow are predicted in a multi-scale scheme. To overcome over-fitting, we construct each convolutional layer followed by a global max-pooling layer of the encoder of DepthNet. Our DepthNet regresses depth map in a full scale as input. The input to the FlowNet includes batches of tensors cascade in channel dimension, including the image pairs I_t^o

and I_t^s, the structure flow f_t^s, and its error map. The outputs of our FlowNet are the scene flow vectors for updating the estimated camera poses and depth maps. Our PoseNet regresses the 6-DoF camera pose, i.e. the 3D translation vector and the 3D Euler angle. The architecture is similar with SfM-Learner,[21] which contains 8 convolutional layers followed by two group fully-connected layers before final prediction. The kernel size is 3 for all convolutional layers except for the first two convolutional layer in two modules where their kernel sizes are 7 and 5, respectively. What's more, the stride of each convolutional layer is 2. We adopt batch normalization [37] and ReLUs [38] interlaced with all the convolutional layers.

4 Experiments and Results

In this section, we evaluate our Un-VDNet with qualitative and quantitative results of visual odometry and depth estimation, and compare those to the state-of-the-art methods on KITTI dataset[12] and TUM dataset.[13] The test time for each sample is comparable to previous works.

4.1 Implementation details

Our network is built on the TensorFlow framework.[39] Though these sub-networks can be trained together in an end-to-end manner, there is no guarantee that the local gradient optimization could get the network to the optimal point.[11] Therefore, we adopt a stage-wise training strategy, reducing the computational consumption and memory cost. Generally speaking, we first train the PoseNet and the DepthNet, and then by fixing their weights, train the FlowNet thereafter. We perform random resizing, cropping, and other color augmentations to prevent over-fitting. We train our Un-VDNet with Adam optimizer, where β_1 = 0.9 and β_2 = 0.99, on eight cores of 3.4 GHZ Intel Core i7-3770 and a NVIDIA TITAN X GPU. The base learning rate is 0.002 and it is decreased manually, and the mini-batch size is 8 for each sub-network. The training typically converges after about 200 epochs with a batch size of 16. For photometric consistency loss, we empirically set α = 0.85. For the pose consistency loss, we set θ = 0.25 on KITTI dataset and 0.75 on TUM dataset verified by experiments. For the loss weights of the final loss function, we empirically find that the combination (λ_p, λ_d, λ_f) = (1.0, 0.1, 10.0) leads to a stable training process.

KITTI dataset[12] contains 61 outdoor video sequences with 42,382 rectified stereo sequences and the image size of 1242 × 375. We resize images into 608 × 160 in our training setup to reduce processing time and computation cost. For visual odometry, we utilize the KITTI$_{odom}$ split[40] which contains 21 sequences. We use Seq.00 - 08 to train PoseNet while the rest of sequences to test like SfM-Learner.[21] During training, we set the sequence length as 5. For depth estimation, we utilize the KITTI$_{eigen}$[12] split, which contains 23,488 stereo images of 33 scenes for training, and 697 images of 28 sequences for testing. We follow this setup and set the sequence length as 3.

TUM dataset[13] contains 89 indoor video sequences taken with three Microsoft Kinect cameras, which divided into 52 sequences for training, 33 sequences for validating, and 4 sequences for testing. We train our Un-VDNet using 52 sequences and test on three sequences including moving persons, such as seq1 (fr2/desk-with-person), seq2 (fr3/sitting-xyz), and seq3 (fr3/walk-xyz). Besides, we resize the images from 640 × 480 to 320 × 240 for reducing processing time and recover them before output following the previous work.[7]

Performance Metric. Following Eigen *et al.*,[12] we adopt the following performance metric:

$$RMS : \sqrt{\frac{1}{T}\sum_{i\in T} ||d_i - d_i^{gt}||^2} \tag{9}$$

$$log\,RMS : \sqrt{\frac{1}{T}\sum_{i\in T}||log\left(d_i\right) - log\left(d_i^{gt}\right)||^2} \tag{10}$$

$$abs.relative : \frac{1}{T}\sum_{i\in T}\frac{d_i - d_i^{gt}}{d_i^{gt}} \tag{11}$$

$$sq.relative : \frac{1}{T}\sum_{i\in T}\frac{||d_i - d_i^{gt}||^2}{d_i^{gt}} \tag{12}$$

$$accuracies : \%\; of\; d_i\; s.t.\; max\left(\frac{d_i}{d_i^{gt}}, \frac{d_i^{gt}}{d_i}\right) = \delta < thr \tag{13}$$

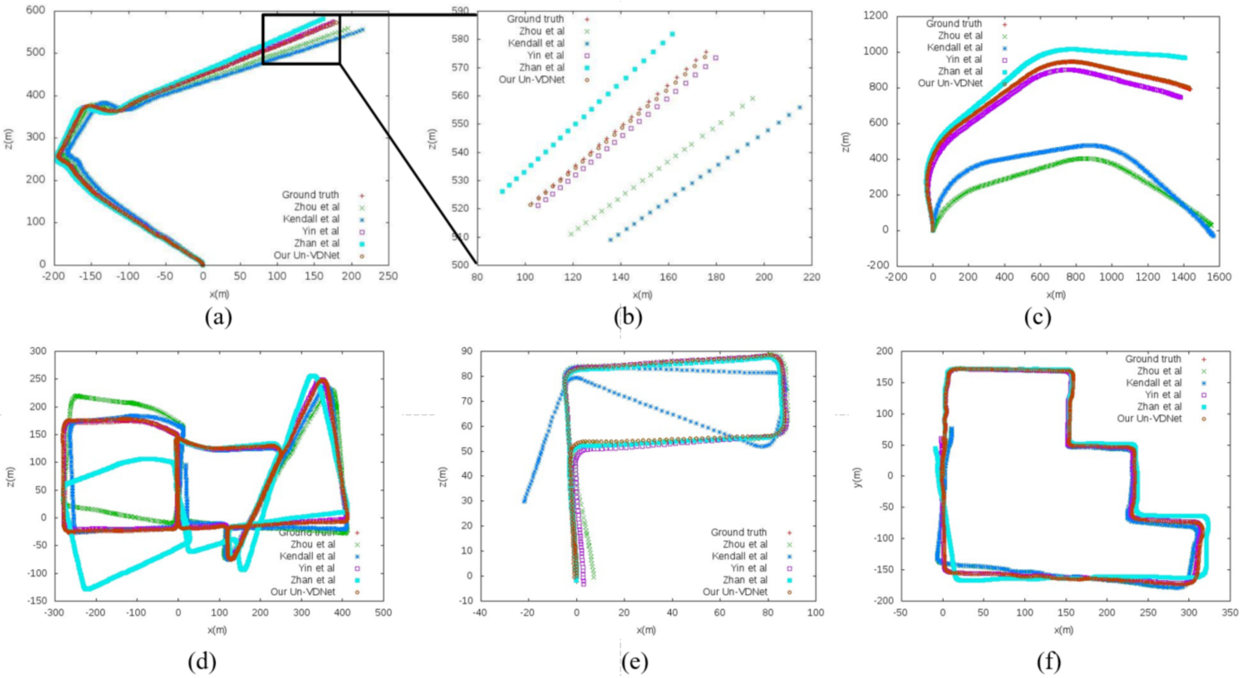
$$ATE_{RMSE} : \sqrt{\frac{1}{T}\sum_{i\in T}||d_i - d_i^{gt}||^2}. \tag{14}$$

where d_i and d_i^{gt} are the predicted depths and ground-truth depth respectively at pixel indexed by i, and T is the total number of pixels in all the evaluated images.

4.2 Results of Visual Odometry

We use the KITTI$_{odom}$ split[40] and TUM dataset[13] mentioned above to evaluated the performance of our proposed PoseNet. The test time is $7ms$ per image. The results on KITTI dataset are compared with Zhou $et\ al.$,[21] Kendall $et\ al.$,[17] Yin $et\ al.$,[11] and Zhan $et\ al.$,[28] for camera motion estimation.

The detailed results of translation and rotation estimation on KITTI dataset are listed in Table 1. Average translation RMSE drift $\mathbf{t_{rel}}$ $(\%)$ and average rotation RMSE drift $\mathbf{r_{rel}}$ $(°/100m)$ on the length of 100m - 800m are adopted. As can be seen from the Table 1, whether it is translation or rotation error, the predictions of our proposed Un-VDNet outperform other four methods which



(a) (b) (c)

(d) (e) (f)

Fig 4 Visual comparison of monocular depth estimation on KITTI dataset.

assume that the scene is a static or don't consider pose consistency constraint. Our $\mathbf{t_{rel}}$ is 4% and $\mathbf{r_{rel}}$ is $1.5°/100m$ lower than the mean translation error and the mean rotation average error rate of others on Seq.09, while 10.9% and $4.0°/100m$ on Seq.10. Because we apply a pose consistency loss to minimize the translation and rotation errors between adjacent frames.

The trajectories of Seq.11 - 15 on KITTI$_{odom}$ split[40] learned from different methods are shown in Figure 4. Figure 4 (a) and (b) about Seq.11 indicates that the estimated ego-motion of the Un-VDNet is the closet to the ground-truth while the Seq.11 trajectory predicted from Kendall $et\ al.$[17] is farthest. Our method yields better visualizations with clearer transitions and consistent local details, especially at corners. For example, at $x_{(m)} = -150$ and $z_{(m)} = 370$ in (a), our predicted camera pose is coincident with the ground-truth. In addition, errors between the estimated trajectories and ground-truth increase over time because of the accumulation of previous prediction errors. However, our results are still the closest to the ground-truth as can be seen from (b).

Table 1 Visual odometry results on Seq.09-10 of KITTI. $\mathbf{t_{rel}}\,(\%)$ and $\mathbf{r_{rel}}\,(^\circ/\mathbf{100m})$ are average translation and rotation RMSE drift on the length of 100 m - 800 m respectively.

Method	Seq. 09		Seq. 10	
	$\mathbf{t_{rel}}$	$\mathbf{r_{rel}}$	$\mathbf{t_{rel}}$	$\mathbf{r_{rel}}$
Zhou *et al.*[21]	15.37	4.06	37.91	17.78
Kendall *et al.*[17]	17.84	6.78	12.45	3.46
Yin *et al.*[11]	12.27	3.78	23.61	4.11
Zhan *et al.*[28]	11.93	3.91	12.95	3.64
Our Un-VDNet	**10.07**	**3.14**	**11.06**	**3.29**

Table 2 Absolute trajectory error (ATE) on TUM dataset. Seq1 is fr2/desk-with-person, seq2 is fr3/sitting-xyz, and seq3 is fr3/walk-xyz.

Method	seq1	seq2	seq3
Zhou *et al.*[21]	1.818	1.228	0.914
Tateno *et al.*[7]	1.927	1.567	0.636
Kendall *et al.*[17]	1.329	1.592	0.706
Yin *et al.*[11]	0.946	1.273	0.682
Zhan *et al.*[28]	1.105	0.981	0.660
Our Un-VDNet	**0.881**	**0.864**	**0.598**

Similarly, Figure 4 (c) - (f) demonstrate that our performance is superior to other methods and robust when the camera is in a pure-rotation station.

Besides, our predicted Absolute Trajectory Error (ATE) is compared with the state-of-the-art methods on TUM dataset[13] as shown in Table 2, which is computed as the root mean square error between the learned camera trajectory and the ground-truth for each tested sequence, fr/desk-with-person, fr3/sitting-xyz, and fr3/walk-xyz. The ATE of our proposed Un-VDNet is the lowest of all three sequences and our mean ATE is 0.38 lower than the average of all other methods on these three sequences, which proves that our network is more efficient and performs better than other methods in indoor scenes.

From Table 1 and 2, we can summarize that camera poses estimated from our network outperform other methods, and a large part of the reason is that we have added a pose consistency loss and constructed a geometric consistency constraint to reduce the errors caused by dynamic objects.

4.3 Results of Depth Estimation

To evaluate the performance of our Un-VDNet in monocular depth estimation, we take the split of KITTI$_{eigen}$[12] and TUM dataset[13] for training. The test time of our proposed network is 10ms for predicting one image depth. The results are compared with Eigen $et\ al.$,[12] Liu $et\ al.$,[42] Garg $et\ al.$,[25] Godard $et\ al.$,[27] Zhan $et\ al.$,[28] and Li $et\ al.$[22] for depth map estimation.

Table 3 reports the performance comparison on the KITTI dataset. We can see the proposed Un-VDNet achieves better depth estimation performance in terms of $abs.$Rel, logRMS, RMS, and threshold accuracy metrics as compared to the state-of-the-art methods. In term of $sq.$ Rel, Un-VDNet outperforms most compared methods and has quite marginal performance degradation as compared to the approaches in Garg $et\ al.$.[25] Figure 5 illustrates some example depth maps produced by our Un-VDNet and the state-of-the-art methods. It can be observed that our proposed framework characterizes the global structure of the scene more precisely and more effectively. Our method predicts better depth map especially in the part of the dynamic car, which presents a clearer contour. We can learn that these depth maps are closet to the ground-truth in the dynamic areas. In addition, we can also estimate the depth for fine objects.

Moreover, Table 4 reports the performance comparison on TUM dataset.[13] Our predicted error metrics are the lowest except RMS and all accuracy metrics are the highest. Therefore, we can observe that the proposed Un-VDNet achieves consistent performance improvements in terms of error and accuracy metrics as compared to the state-of-the art methods. Figure 6 illustrates some

Table 3 The error and accuracy metrics of depth estimation on KITTI dataset.

Method	Error				Accuracy		
	abs. Rel.	sq. Rel.	log RMS	RMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.*[12]	0.203	1.548	0.282	6.307	0.702	0.890	0.958
Liu *et al.*[42]	0.201	1.584	0.273	6.471	0.680	0.898	0.967
Garg *et al.*[25]	0.152	**1.226**	0.246	5.849	0.784	0.921	0.967
Godard *et al.*[27]	0.148	1.344	0.247	5.927	0.803	0.922	0.964
Zhan *et al.*[28]	**0.144**	1.391	0.241	5.869	0.803	0.928	0.969
Li *et al.*[22]	0.183	1.730	0.268	6.570	0.691	0.902	0.968
Our Un-VDNet	**0.144**	1.312	**0.242**	**5.827**	**0.816**	**0.931**	**0.970**

Table 4 The error and accuracy metrics of depth estimation on TUM dataset.

Method	Error				Accuracy		
	abs. Rel.	sq. Rel.	log RMS	RMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.*[12]	1.709	0.270	0.183	1.159	0.734	0.902	0.959
Liu *et al.*[42]	1.565	0.275	0.198	1.836	0.718	0.901	0.960
Garg *et al.*[25]	0.737	0.232	0.153	1.328	0.802	0.934	0.972
Godard *et al.*[27]	0.541	0.273	0.169	**1.080**	0.740	0.904	0.962
Zhan *et al.*[28]	0.698	0.221	0.143	1.162	0.810	0.947	0.982
Li *et al.*[22]	0.857	0.233	0.155	1.296	0.793	0.931	0.973
Our Un-VDNet	**0.401**	**0.186**	**0.132**	1.097	**0.853**	**0.971**	**0.995**

example depth maps produced by the Un-VDNet on three dynamic scenes of TUM dataset.[13] We can see that the error of depth map estimated from our UN-VDNet is the lowest especially in the part of moving persons, compared to the ground-truth. Especially for the moving body or arms, our proposed method can clearly predict the depth and has a more complete outlines. It can be drawn that the geometric consistency constraint combined with photometric consistency contributes to the performance of our network and makes it more robust to the dynamic environment.

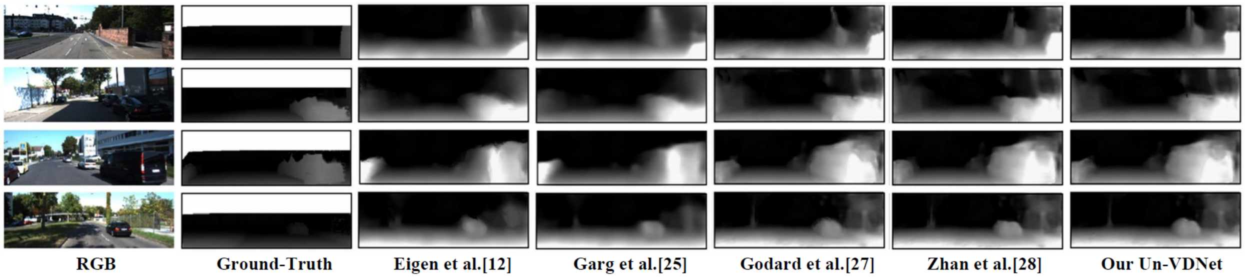


Fig 5 Visual comparison of monocular depth estimation on KITTI dataset.

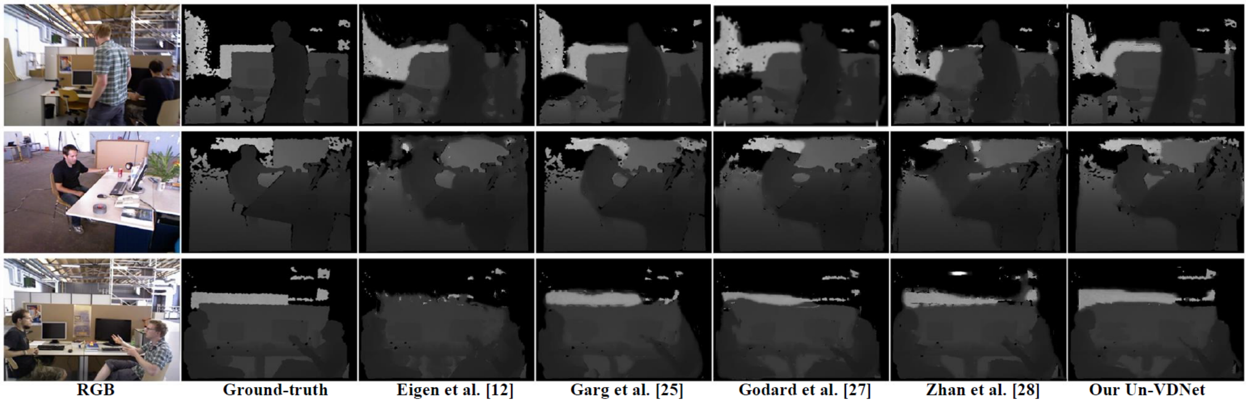


Fig 6 Visual comparison of monocular depth estimation on TUM dataset.

4.4 Ablation Studies

We conducted ablation experiments on KITTI and TUM dataset to investigate the effectiveness of our proposed Un-VDNet.

Effectiveness of Pose Consistency. Table 5 reports the performance in visual odometry of the Un-VDNet and its several variants on KITTI and TUM dataset. Un-VDNet $\setminus$ P refers to the Un-VDNet without the use of pose consistency constraint. From the Table 5, we can obtain the following observations: Un-VDNet $\setminus$ P results in heave performance degradation in absolute trajectory error (ATE) compared to the Un-VDNet. The ATE of Un-VDNet is 34$\%$ lower than Un-VDNet $\setminus$ P on KITTI dataset, and 23$\%$ on TUM dataset. It demonstrates that the pose consistency constraint is useful for improving camera ego-motion estimation and leads to more accurate trajectories.

Table 5 Evaluation of the effectiveness of pose consistency of ATE in visual odometry within the Un-VDNet. "K" means KITTI and "T" means TUM dataset.

Method	Dataset	Seq.09	Seq.10	Seq.1	Seq.2	Seq.3
Un-VDNet $\setminus$ P	K	0.020 ± 0.006	0.029 ± 0.007	—	—	—
Un-VDNet	K	0.012 ± 0.008	0.013 ± 0.009	—	—	—
Un-VDNet $\setminus$ P	T	—	—	1.132	1.147	0.760
Un-VDNet	T	—	—	0.881	0.846	0.598

Table 6 Evaluation of the effectiveness of flow estimation in depth maps within the Un-VDNet on KITTI dataset.

Method	Error				Accuracy		
	$abs.$ Rel.	$sq.$ Rel.	log RMS	RMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Un-VDNet $\setminus$ F	0.173	1.439	0.261	5.890	0.645	0.881	0.959
Un-VDNet	0.144	1.312	0.242	5.827	0.816	0.931	0.970

Effectiveness of Flow Estimation. Table 6 reports the performance comparison on KITTI dataset between our proposed Un-VDNet network and the similar model without the FlowNet module. Un-VDNet $\setminus$ F refers to the model without flow estimation. From the Table 6, we can see that the Un-VDNet with flow estimation constrained by geometric consistency achieves better performance improvements in terms of all error and accuracy metrics. For example, the Un-VDNet obtains only 0.144 $abs.$Rel. and achieves 16.76% error decrease as compared to the Un-VDNet $\setminus$ F. In terms of the threshold accuracy with $\delta < 1.25$, it achieves 0.816 accuracy which is 26.51% better than the model without flow estimation. These results demonstrate that the network with flow estimation is able to boost depth estimation by geometric consistency constraint.

5 Conclusion

In this paper, we propose a novel framework, Un-VDNet, based on unsupervised CNNs for monocular visual odometry and depth estimation between outdoor and indoor scenes. We train the entire framework with stereo sequences and test on monocular sequences based on TensorFlow. Different from previous CNN-based networks which assume that the scene is static, our network considers dynamic objects additionally. The PoseNet and the DepthNet learn 6-DOF camera pose and depth map with photometric consistency constraints firstly, and then we train the FlowNet with a geometric consistency loss to penalize the error between the scene optical flow and the structure flow generated by predicted pose and depth. Considering the dynamic scenes, the proposed geometric estimator, FlowNet, enhances the performance of the structure estimator, PoseNet and DepthNet. Moreover, we also propose a novel pose consistency to constrain the estimated translation and rotation vectors of frame-to-frame odometry without scale ambiguity. The experimental results show that Un-VDNet outperforms state-of-the-art methods in pose and depth estimation at pixel-level without costly ground-truth, and it is more effective in dealing with dynamic scenes compared with previous methods on KITTI dataset and TUM dataset.

The main limitation of our approach is that it doesn't exploit any semantic information, which can be explored in the future work. We would like to introduce semantic estimation into our framework and reconstruct the scene structure with 3D and semantic cues. Another potential research direction is to apply our depth estimation model to traditional visual tasks and benefit them, such as object detection, segmentation, and so on.

Acknowledgments

The work presented in this paper was supported by Beijing Natural Science Foundation of China (Grant No. L182033), Fund for Beijing University of Posts and Telecommunications (2019PTB-001).

References

1 F. Fraundorfer, C. Engels, and D. Nistér, “Topological mapping, localization and navigation using image collections,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3872–3877, IEEE (2007).

2 X. Yang, J. Chen, Z. Wang, *et al.*, “Monocular camera based real-time dense mapping using generative adversarial network,” in *ACM International Conference on Multimedia*, 896–904, ACM (2018).

3 A. Pumarola, A. Vakhitov, A. Agudo, *et al.*, “Pl-slam: Real-time monocular visual slam with points and lines,” in *IEEE International Conference on Robotics and Automation*, 4503–4508, IEEE (2017).

4 R. Valencia and J. Andrade-Cetto, “Active pose slam,” in *Mapping, Planning and Exploration with Pose SLAM*, 89–108, Springer (2018).

5 G. Zhang, H. Liu, Z. Dong, *et al.*, “Efficient non-consecutive feature tracking for robust structure-from-motion,” *IEEE Transactions on Image Processing* **25**(12), 5957–5970 (2016).

6 C. Toft, E. Stenborg, L. Hammarstrand, *et al.*, “Semantic match consistency for long-term visual localization,” in *European Conference on Computer Vision*, 383–399, Springer (2018).

7 K. Tateno, F. Tombari, I. Laina, *et al.*, “Cnn-slam: Real-time dense monocular slam with learned depth prediction,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 6243–6252, IEEE (2017).

8 T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual slam algorithms: A survey from 2010 to 2016,” *IPSJ Transactions on Computer Vision and Applications* **9**(1), 16 (2017).

9 L.-C. Chen, G. Papandreou, I. Kokkinos, *et al.*, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(4), 834–848 (2018).

10 B. Ummenhofer, H. Zhou, J. Uhrig, *et al.*, “Demon: Depth and motion network for learning monocular stereo,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 5038–5047, IEEE (2017).

11 Z. Yin and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1983–1992, IEEE (2018).

12 D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in Neural Information Processing Systems*, 2366–2374, Curran Associates, Inc. (2014).

13 J. Sturm, N. Engelhard, F. Endres, *et al.*, “A benchmark for the evaluation of rgb-d slam systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 573–580, IEEE (2012).

14 D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry,” in *IEEE Conference on Computer Vision and Pattern Recognition*, I–I, IEEE (2004).

15 S. M. Seitz, B. Curless, J. Diebel, *et al.*, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 519–528, IEEE (2006).

16 J. Zbontar, Y. LeCun, *et al.*, “Stereo matching by training a convolutional neural network to compare image patches,” *Journal of Machine Learning Research* **17**(1), 2287–2318 (2016).

17 A. Kendall, M. Grimes, and R. Cipolla, “Posenet: A convolutional network for real-time 6-dof camera relocalization,” in *IEEE International Conference on Computer Vision*, 2938–2946, IEEE (2015).

18 G. Costante, M. Mancini, P. Valigi, *et al.*, “Exploring representation learning with cnns for frame-to-frame ego-motion estimation,” *IEEE Robotics and Automation Letters* **1**(1), 18–25 (2016).

19 A. Handa, M. Bloesch, V. Pătrăucean, *et al.*, “Gvnn: Neural network library for geometric computer vision,” in *European Conference on Computer Vision*, 67–82, Springer (2016).

20 J. Wu, L. Ma, and X. Hu, “Delving deeper into convolutional neural networks for camera relocalization,” in *IEEE International Conference on Robotics and Automation*, 5644–5651, IEEE (2017).

21 T. Zhou, M. Brown, N. Snavely, *et al.*, “Unsupervised learning of depth and ego-motion from video,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1851–1858, IEEE (2017).

22 R. Li, S. Wang, Z. Long, *et al.*, “Undeepvo: Monocular visual odometry through unsupervised deep learning,” in *IEEE International Conference on Robotics and Automation*, 7286–7291, IEEE (2018).

23 C. Sweeney, T. Sattler, T. Hollerer, *et al.*, “Optimizing the viewing graph for structure-from-motion,” in *IEEE International Conference on Computer Vision*, 801–809 (2015).

24 K. He, X. Zhang, S. Ren, *et al.*, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778, IEEE (2016).

25 R. Garg, V. K. BG, G. Carneiro, *et al.*, “Unsupervised cnn for single view depth estimation:

Geometry to the rescue," in *European Conference on Computer Vision*, 740–756, Springer (2016).

26 S. Vijayanarasimhan, S. Ricco, C. Schmid, *et al.*, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804* (2017).

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

28 H. Zhan, R. Garg, C. S. Weerasekera, *et al.*, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 340–349, IEEE (2018).

29 R. Pio, "Euler angle transformations," *IEEE Transactions on Automatic Control* **11**(4), 707–715 (1966).

30 M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2017–2025, Curran Associates, Inc. (2015).

31 H. Zhao, O. Gallo, I. Frosio, *et al.*, "Loss functions for neural networks for image processing," *arXiv preprint arXiv:1511.08861* (2015).

32 Z. Wang, A. C. Bovik, H. R. Sheikh, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004).

33 L. Zhang, L. Zhang, X. Mou, *et al.*, "A comprehensive evaluation of full reference image quality assessment algorithms," in *IEEE International Conference on Image Processing*, 1477–1480, IEEE (2012).

34 N. Mayer, E. Ilg, P. Hausser, *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 4040–4048, IEEE (2016).

35 E. Ilg, N. Mayer, T. Saikia, *et al.*, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2462–2470, IEEE (2017).

36 A. Dosovitskiy, P. Fischer, E. Ilg, *et al.*, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2758–2766, IEEE (2015).

37 S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167* (2015).

38 V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 807–814, IMLS (2010).

39 M. Abadi, P. Barham, J. Chen, *et al.*, "Tensorflow: a system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283, USENIX Association (2016).

40 A. Geiger, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361, IEEE (2012).

41 D. Xu, E. Ricci, W. Ouyang, *et al.*, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 5354–5362, IEEE (2017).

42 F. Liu, C. Shen, G. Lin, *et al.*, "Learning depth from single monocular images using deep

convolutional neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(10), 2024–2039 (2015).

43 O. H. Jafari, O. Groth, A. Kirillov, *et al.*, "Analyzing modular cnn architectures for joint depth prediction and semantic segmentation," in *IEEE International Conference on Robotics and Automation*, 4620–4627, IEEE (2017).

44 I. Laina, C. Rupprecht, V. Belagiannis, *et al.*, "Deeper depth prediction with fully convolutional residual networks," in *IEEE International Conference on 3D Vision (3DV)*, 239–248, IEEE (2016).

Xuyang Meng is a Ph.D student at Beijing University of Posts and Telecommunications. She received her BS degree in engineering from Yanshan University in 2016, and her MS degree in engineering from Beijing University of Posts and Telecommunications in 2019.

Chunxiao Fan is currently a professor and director of Center for information electronic and intelligence system. She served as a member of Chinese Sensor network working group. She was elevated to evaluation expert of Beijing Scientific and Technical Academy Awards. Her research interests include Heterogeneous media data analysis, Internet of Things, data mining, communication software. She has published more than 30 papers in international journals and conferences, authored and edited three books and authorized several invention patents.

Yue Ming received the Ph.D degree in Signal and Information Processing from Beijing Jiaotong University, China, in 2013. She worked as a visiting scholar in Carnegie Mellon University, U.S., between 2010 and 2011. Since 2013, she has been working as a faculty member at Beijing University of Posts and Telecommunications. Her research interests are in the areas of biometrics, computer vision, computer graphics, information retrieval, pattern recognition, etc. She has authored more than 40 scientific papers.

Yuan Shen received the Ph.D. degree in 2014, from Beijing Jiaotong University, Beijing, China. His research interests include Machine Learning, Deep Learning, objects detection and segmentation, multi-object tracking, and trajectory analysis. Now, he is working on autonomous driving for environmental perception in Autonomous Driving Center, Tencent Technology (Beijing) Co., Ltd.

Hui Yu is Professor with the University of Portsmouth, UK. Prof. He used to work at the University of Glasgow before moving to the University of Portsmouth. His research interests include methods and practical development in vision, machine learning and AI with applications to human-machine interaction, Virtual and Augmented reality, robotics and geometric processing of facial expression. He serves as an Associate Editor of IEEE Transactions on Human-Machine Systems and Neurocomputing journal.

List of Figures

1 Illustration of our proposed framework Un-VDNet. The network consists of three sub-networks, PoseNet. DepthNet, and FlowNet. It is trained on unlabeled RGB images in unsupervised manner and estimates the camera pose T(t,r) and image depth, where T $\in \mathbb{R}^{4\times4}$ is transform matrix, t $\in \mathbb{R}^3$ is translation, and r $\in \mathbb{R}^3$ is rotation.

2 Architecture of our proposed framework, Un-VDNet. The framework consists of three sub-networks: a PoseNet for camera pose, a DepthNet for depth map, and a FlowNet for flow estimation. Details are described in Section 3.

3 The process of left image warping. The DepthNet predicts inverse depth from left image L. Then we use Spatial Transformer (STN) [30] to synthesize a new right image R’ with inverse depth map and raw right image R. In STN, X and Y are feature maps, TP is transformation parameters, and G is sampling grid. We can synthesize a new left image L’ in the same way.

4 Visual comparison of monocular depth estimation on KITTI dataset.

5 Visual comparison of monocular depth estimation on KITTI dataset.

6 Visual comparison of monocular depth estimation on TUM dataset.

List of Tables

1 Visual odometry results on Seq.09-10 of KITTI. $\mathbf{t_{rel}}\,(\%)$ and $\mathbf{r_{rel}}\,(^\circ/\mathbf{100m})$ are average translation and rotation RMSE drift on the length of 100 m - 800 m respectively.

2 Absolute trajectory error (ATE) on TUM dataset. Seq1 is fr2/desk-with-person, seq2 is fr3/sitting-xyz, and seq3 is fr3/walk-xyz.

3 The error and accuracy metrics of depth estimation on KITTI dataset.

4 The error and accuracy metrics of depth estimation on TUM dataset.

5 Evaluation of the effectiveness of pose consistency of ATE in visual odometry within the Un-VDNet. “K” means KITTI and “T” means TUM dataset.

6 Evaluation of the effectiveness of flow estimation in depth maps within the Un-VDNet on KITTI dataset.