3D Hierarchical Refinement and Augmentation for Unsupervised Learning of Depth and Pose From Monocular Video

Guangming Wang, Graduate Student Member, IEEE, Jiquan Zhong, Shijie Zhao, Wenhua Wu, Zhe Liu, and Hesheng Wang, Senior Member, IEEE

Abstract—Depth and ego-motion estimations are essential for the localization and navigation of autonomous robots and autonomous driving. Recent studies make it possible to learn the per-pixel depth and ego-motion from the unlabeled monocular video. In this paper, a novel unsupervised training framework is proposed with 3D hierarchical refinement and augmentation using explicit 3D geometry. In this framework, the depth and pose estimations are hierarchically and mutually coupled to refine the estimated pose layer by layer. The intermediate view image is proposed and synthesized by warping the pixels in an image with the estimated depth and coarse pose. Then, the residual pose transformation can be estimated from the new view image and the image of the adjacent frame to refine the coarse pose. The iterative refinement is implemented in a differentiable manner in this paper, making the whole framework optimized uniformly. Meanwhile, a new image augmentation method is proposed for the pose estimation by synthesizing a new view image, which creatively augments the pose in 3D space but gets a new augmented 2D image. The experiments on KITTI demonstrate that our depth estimation achieves state-of-the-art performance and even surpasses recent approaches that utilize other auxiliary tasks. Our visual odometry outperforms all recent unsupervised monocular learning-based methods and achieves competitive performance to the geometry-based method, ORB-SLAM2 with back-end optimization. The source codes will be released soon at: https://github.com/IRMVLab/HRANet.

Index Terms—Monocular depth estimation, visual odometry, unsupervised learning, pose refinement, 3D augmentation.

I. INTRODUCTION

T

HE depth and ego-motion (which also can be noted as pose) estimations [1], [2], [3] are the basis of autonomous robot localization and navigation [4], [5], [6], [7]. The traditional monocular Simultaneous Localization and Mapping (SLAM) system needs initialization. The manual feature extraction is difficult to generalize in various environments, such as the environment with less texture. Dense depth estimation [8], [9] based on deep learning has been developed with superior performance from only monocular camera. However, the ground truth of dense depth is expensive. Recently, unsupervised learning enables the depth and pose jointly learned without labels [1], [2], [10], [11], [12], [13], [14]. After network training, depth network and pose network can also be evaluated separately so that depth estimation no longer relies on the existing pose estimation, and 3D information can be obtained only from a monocular image. Moreover, image reconstruction is a key step in unsupervised joint learning of depth and pose. However, due to the occlusion, dynamics, and illumination variation, the image reconstruction is not always perfect. Many studies focus on the loss calculation and make the reconstruction error influenced less by the occlusion or dynamic objects in the process of network training [12], [13], [14], [15]. Some works exploit joint learning with other auxiliary tasks to handle this challenge [16], [17], [18], [19], [20], [21].

In this paper, we propose a new perspective to improve the performance of joint learning for depth and pose estimation. Specifically, the pose estimation tends to degrade due to the considerable distance between two adjacent frames and the influence of the occlusion area. Therefore, we generate new intermediate views to enhance the accuracy of pose estimation, as shown in Fig. 1. Due to the joint training, the depth network is also affected and improved simultaneously. Meanwhile, the pose estimated by the coarse pose network can transform the predicted 3D point cloud in one frame to an adjacent frame based on the depth estimation results. Then, we generate masks from the operation above to solve the problem of occlusion and dynamic objects. After the coarse pose transformation, the point cloud is reprojected onto the image plane of the adjacent frame, which makes it easier to match the pixels of adjacent frames. It is worth mentioning that the depth network has multi-scale depth outputs, and the method proposed in...
Fig. 1. The figure shows the process of pose estimation refinement. For details, given two adjacent frames $X^t$ and $X^{t+1}$, with their corresponding camera view $P^t$ and $P^{t+1}$, we utilize our 3D hierarchical refinement method introduced in Sec. III-B to synthesize transitional camera view $D.T^t$ and get residual pose $T^t_m$ ($m = 1, 2, \ldots, M - 1$). By multiplying $T_1, T_1^t \cdots T_{M-1}^t$ step by step, we could refine our pose continuously.

this paper also has multiple pose estimations. Therefore, there are many combinations to associate them and construct joint constraints. The best way is demonstrated through experiments and corresponding analysis in this paper.

In addition, the commonly used KITTI dataset [22] has only sample camera motion patterns [23], which makes the trained pose network less generalized. Wang et al. [23] constructed a new dataset with complex and diverse motion patterns and changeable scenes. However, the dataset is virtual, and the domain gap is unavoidable [24]. It is costly to simultaneously meet the complex and diverse motion patterns of the environment in practice. To improve the generalization capability of the pose network, diverse motion patterns are required during training. Therefore, a new 6 Degrees-of-Freedom (DoF) pose augmentation method is proposed based on the depth estimation in the joint unsupervised training to synthesize new views from raw images, thereby enriching the training data and improving the performance of the pose network.

In general, the contributions of this paper are as follows:

- Relying on the explicit 3D information from depth estimation, the concept of intermediate view synthesis is proposed in the joint unsupervised learning of depth and pose. A new joint unsupervised learning framework of depth and pose with the iterative view synthesis and pose refinement is built to refine the estimated pose in an end-to-end fashion.

- In the proposed framework, 6-DoF pose augmentation is proposed to synthesize a new view image, making the dataset for the training cheaply expanded. To our best knowledge, this paper is the first to do the 6-DoF pose augmentation in visual odometry learning. The whole framework is implemented in a differentiable manner to optimize the depth and pose jointly.

- The mutual coupling learning of multi-scale depth estimation and multiple pose estimation is carefully considered and demonstrated by experiments on the KITTI dataset, which prove the proposed method achieves state-of-the-art performance on both depth and pose estimations. Besides, without relying on other auxiliary tasks and postprocessing, our visual odometry achieves competitive performance to the geometry-based method, ORB-SLAM2 [25] with back-end optimization and closed-loop optimization.

II. RELATED WORK

A. Depth Supervised Learning

Learning scene depth from a single image by Convolutional Neural Network (CNN) can date back to Eigen et al. [8]. They use two deep neural networks for depth estimation of images, one for global estimation and the other for refinement of local estimation. Liu et al. [26] express depth estimation as a continuous Conditional Random Field (CRF) problem and propose a deep convolution neural field model. Kuznietsov et al. [27] propose a monocular depth map prediction method based on semi-supervised learning using sparse ground truth depth. Guo et al. [28] solve the gap between synthetic data and real data by using stereo matching as a proxy task.

B. Depth Self-Supervised Learning From Stereo Video

Garg et al. [29] propose the first end-to-end unsupervised convolutional neural network for depth estimation, which eliminates the need for large amounts of labeled data and greatly reduces the cost of training. As image reconstruction alone leads to low-quality depth maps, Godard et al. [30] propose a new kind of training loss that enhances consistency between the left and right images. Zhan et al. [31] consider the information of consecutive frames and propose an additional feature reconstruction loss. SGANVO [32] proposes a novel pose network that uses the stacked generative adversarial structure and gets good performance due to its recurrent module across the network. Wong et al. [33] propose an adaptive weight to adjust the proportion of regularization loss and a bilateral cyclic consistency loss to solve the occlusion problem. They also built a two-branch decoder, with one branch predicting the initial disparity and the other refining it. EffiScene [34] proposes per-pixel rigidity to perform accurate rigid constraint on joint learning of optical flow, depth, and camera pose.

C. Depth Self-Supervised Learning From Monocular Video

Zhou et al. [10] firstly propose a self-supervised method for depth together with pose estimation by minimizing the pixel-based loss only from monocular video. Many studies for self-supervised learning of monocular depth and pose are based on eliminating the influence of the occlusion, dynamics, and illumination variation when reconstructing the image between adjacent frames. To solve these problems, GeoNet [17] proposes jointly learning for depth, pose and optical flow and use forward optical flow and backward optical flow to mask occlusion areas. DF-Net [18] uses depth and pose to get static optical flow, then uses geometry consistency to mask occlusion with estimated static optical flow. They also use the optical flow estimated by the optical flow network and the static flow obtained by the depth-pose networks to train the depth, pose and optical flow tasks. CC [19] and EPC++ [35] train a motion segmentation net to split static
areas for depth and pose learning and dynamic areas for optical flow learning. Monodepth2 [14] uses the minimum photometric error in multiple source frames for each pixel, which most likely does not contain the occlusion and dynamics information. SC-SFM [13] masks occluded pixels in images by computing the depth inconsistency probability map and proposes the depth consistency. Gordon et al. [15] learn the camera’s parameters from the video to make the depth prediction more accurate, and use the predicted depth map directly to deal with occlusion. DOP [20] divides an image into three regions, static regions, dynamic regions, and occlusion regions by the information of adjacent frames. They solve the occlusion problem by explicit geometric calculation using the predicted point cloud. D3VO [36] predicts transform parameters to align the brightness condition to reduce the impact of illumination variation. LEGO [37] estimates edge and 3D information together to improve the accuracy of detail estimation. Shen et al. [38] introduce geometric constraint matching loss to compensate for the limitation of unsupervised loss of depth and pose estimation. Mahjourian et al. [39] construct the 3D point clouds from estimated depth maps and propose a new 3D Iterative Closest Point (ICP) loss.

A better network structure can improve the performance of the depth and pose estimation. CM-VO [40] calculates the confidence of poses and refines them. GANVO [11] uses a generator and discriminator structure to optimize the reconstructed image. Wang et al. [41] propose a differentiable implementation of DVO (DDVO module) that takes full advantage of the relationship between camera pose and depth prediction. Zhao et al. [42] use Generative Adversarial Networks (GAN) to generate real images and virtual images, and propose the depth consistency loss between the depth estimated from virtual image and the depth estimated from real image, which improves the performance of depth estimation network. Li et al. [43] propose a new residual translation field regularization method. They decompose the translation field into background translation and object translation with respect to the background.

### D. Novel View Synthesis

Novel view synthesis is used in many computer vision complications like video interpolation [44] and self-supervised optical flow learning [45]. Depth-Image-Based-Rendering (DIBR) techniques synthesize the novel views by the means of reference images and corresponding depth maps. Due to the numerical rounding-off error in forward warping process, inaccurate depth, and the lack of background information, DIBR views contain many artifacts like empty and translucent cracks, ghosts, disocclusions, and out-of-field areas. In order to solve these problems, Ahn et al. [46] fill the cracks by the median filter. Oliveira et al. [47], [48] detect the cracks by shape analysis and fill them through the hierarchical hole-filling (HHF) algorithm [49]. After detecting the ghost areas, Oliveira et al. [47] and Liang et al. [50] correct the disparity values of the ghost candidate pixels, and Ahn et al. [46] remove the corresponding areas in the synthesized view. The disocclusions and out-of-field areas usually are inpainted by patch matching [51] with patch priorities. Many methods [46], [47], [48], [50], [52], [53] propose effective priorities computing methods and obtain good performance. Based on the great representation capacities of neural networks, the latest NeRF [54] and GRF [55] use neural radiance fields to represent 3D scenes and get impressive results in novel view synthesis. However, these methods [54], [55] need extra networks for image feature extracting and neural rendering.

Different from the above, we focus on a new pose refinement method based on end-to-end fashion and 6-DoF pose augmentation to realize robustness and superiority. To the best of our knowledge, this paper is the first to consider and realize 6-DoF pose augmentation for visual odometry.

### III. MAIN APPROACH

#### A. Method Overview

Our method aims to promote the unsupervised learning performance of monocular depth and pose only from monocular video. As the pose estimation is critical to depth training, we propose the 3D hierarchical refinement of pose to improve the depth and pose estimation performance. In the proposed 3D hierarchical refinement architecture, DepthNet $N_D$ is used twice for two monocular input images, $X^t$ and $X^{t+1}$, to estimate the monocular depth maps, $D^t$ and $D^{t+1}$, respectively, in time sequences. The PoseNets $N_P$ are used in series. The pose $T_1$ is estimated by the first PoseNet $N_P$.

The following PoseNets $N_{P2}, N_{P3}$...estimate the residual pose transformation to realize the hierarchical refinement of the estimated pose. However, the pose transformation is for 3D points, and the input images to the PoseNets are 2D data. Therefore, the transformed 2D images are required as the intermediate state to realize the residual pose learning. The joint unsupervised learning of depth and pose provides this condition. The estimated depth is used based on 2D-3D transformation to obtain the hierarchical intermediate state.

Through the image reconstruction loss $L^R$, geometry consistency loss $L^G$, and depth smooth loss $L^S$, the multi-scale depths and multi-layer poses are trained together. The three losses will be described detailly in Sec. III-B. As the KITTI dataset [22] has an unbalanced and fixed pose range [23], in order to expand the data distribution efficiently, we propose the data augmentation loss $L^A$ based on 2D-3D transformation in Sec. III-C.

The overall training loss is:

$$L = a_1L^R + a_2L^G + a_3L^S + a_4L^A,$$

where $a_1, a_2, a_3$, and $a_4$ are hyperparameters.

#### B. 3D Hierarchical Refinement With Multi-Scale Depths and Multiple Poses

Our DepthNet $N_D$ has multi-scale depth estimation results, $D_1, D_2, \ldots, D_N$, where $D_1$ means the minimum scale of depth map. Our method has multiple PoseNets, $N_{P1}, N_{P2}, \ldots, N_{PM}$, with the same network structure to refine the pose estimation. The idea is introduced in Fig. 1. $X^t$ is obtained when the robot is at $P^t$, while $X^{t+1}$ is obtained at $P^{t+1}$. The first PoseNet $N_{P1}$ estimates the coarsest pose $T_1$. Then, we use the image warping based on 2D-3D
transformation to construct the intermediate image between \( X^t \) and \( X^{t+1} \) to make next residual pose estimation easier.

1) Intermediate View Synthesis and Pose Refinement: We construct an image \( X_1^{\text{warp}} \) of a virtual intermediate view based on the estimated coarsest pose \( T_1 \) as following 2D-3D transformation:

\[
D_1^{\text{pro}}(i_{\text{pro}}, j_{\text{pro}}) \cdot [i_{\text{pro}}, j_{\text{pro}}, 1]^T = KT_1(D_1^{i+1}(i^{t+1}, j^{t+1}) - K^{-1}[i^{t+1}, j^{t+1}, 1]^T),
\]

where \( K \) is the intrinsic matrix. Formula (2) is the process of finding corresponding coordinates between image \( X_1^{\text{warp}} \) and \( X^t \). The process contains three steps: (a) the 2D pixel coordinates \((i^{t+1}, j^{t+1})\) are projected to 3D camera space coordinates \([X_{t+1}, Y_{t+1}, Z_{t+1}]^T\) with the depth \( D^{t+1} \) and the inverse intrinsic \( K^{-1} \) through the pinhole camera model:

\[
[X_{t+1}, Y_{t+1}, Z_{t+1}]^T = D^{t+1}(i^{t+1}, j^{t+1}), K^{-1}[i^{t+1}, j^{t+1}, 1]^T,
\]

(b) the 3D coordinates \([X_{t+1}, Y_{t+1}, Z_{t+1}]^T\) are transformed to \([X', Y', Z']^T\) with the pose estimation \( T_1\):

\[
[X', Y', Z']^T = T_1[X_{t+1}, Y_{t+1}, Z_{t+1}, 1]^T,
\]

(c) the 3D camera space coordinates \([X', Y', Z']^T\) are projected to 2D pixel coordinates \((i_{\text{pro}}, j_{\text{pro}})\) with intrinsic \( K \) through the pinhole camera model:

\[
D_1^{\text{pro}}(i_{\text{pro}}, j_{\text{pro}}) \cdot [i_{\text{pro}}, j_{\text{pro}}, 1]^T = K[X', Y', Z']^T,
\]

where \((i_{\text{pro}}, j_{\text{pro}})\) are the projected coordinates in \( X_1^{\text{warp}} \) image plane from \((i^{t+1}, j^{t+1})\). Formula (3) is the bilinear interpolation process to obtain the virtual image \( X_1^{\text{warp}} \) from \( X^t \). \((t, l), (t, r), (b, l), (b, r)\) are four adjacent integer pixel coordinates around \((i_{\text{pro}}, j_{\text{pro}})\) \( t \) means top, \( b \) means bottom, \( l \) means left, and \( r \) means right, \( w_{ij} \) are the bilinear interpolation weights and the sum of \( w_{ij} \) for the four adjacent pixel equals 1. \( X_1^{\text{warp}} \) is a virtual image perceived at the view of \( P_1^{\text{warp}} = T_1P_t \), which is a closer view to \( P_t \) than camera view \( P_t \) as shown in Fig. 1.

As shown in Fig. 2, new \( X_1^{\text{warp}} \) and \( X^{t+1} \) are input to the second PoseNet \( N_{P_2} \) and obtain the residual pose estimation \( T'_2 \). The refined pose estimation between two adjacent raw images is calculated by:

\[
T_2 = T'_2 T_1 = N_{P_2}(X_1^{\text{warp}}, X^{t+1})T_1.
\]

Through the refined pose, the new intermediate view picture can be synthesized and input to the next PoseNet combined with \( X^{t+1} \) for the next pose refining. All these calculations are differentiable so that we can optimize the whole system end-to-end.

2) Loss Functions With Masks: Finally, our method has multi-layer pose estimation results, \( T_1, T_2, \ldots, T_M, \ldots, T_M \), where \( T_1 \) means the coarsest pose estimation result. The depth estimation results \( D_1, D_2, \ldots, D_n, \ldots, D_M \) are multi-scale, where \( D_1 \) is the lowest scale. So that, there are a number of combinations of the depths and poses to calculate \( X_1^{\text{warp}} \) in the formula (2) and (3). \( X_{m,n}^{\text{warp}} \) denoted the warped image obtained by the formula (2) and (3) with \( T_m \) and \( D_n \). The warped image \( X_{m,n}^{\text{warp}} \) and raw \( X^{t+1} \) are used to calculate image reconstruction loss \( \sigma(X_{m,n}^{\text{warp}}, X^{t+1}) \) to constrain the training of depth and pose networks to realize the unsupervised learning.
of depth and pose:

$$\sigma(X_{m,n}^{\text{warp}}, X^{t+1}) = \lambda_D \left\| X_{m,n}^{\text{warp}} - X^{t+1} \right\|_1 + (1 - \lambda_D) \frac{1 - \text{SSIM}(X_{m,n}^{\text{warp}}, X^{t+1})}{2},$$  

(8)

where $\lambda_D$ is a hyperparameter. $\text{SSIM}(X_{m,n}^{\text{warp}}, X^{t+1})$ is the structural similarity (SSIM) [56] loss function for image reconstruction with neural networks against illumination changes, which is widely used in [11], [12], [13], and [14]. However, due to occlusion, there are some pixels that cannot be constructed from the adjacent frames, which produces some vacancy in $X_{m,n}^{\text{warp}}$. The depth inconsistency map [13] illustrates the normalized difference between depth $D^t$ and $D^{t+1}$ in each pixel. Some artifacts like occlusions, dynamic objects, and inaccurate depth estimation will lead to the depth inconsistency. Specifically, the depth inconsistency map is calculated by:

$$D_{m,n}^{\text{diff}}(i^{t+1}, j^{t+1}) = \frac{|D_{m,n}^{\text{pro}}(i^{t+1}, j^{t+1}) - D_{m,n}^{\text{warp}}(i^{t+1}, j^{t+1})|}{D_{m,n}^{\text{pro}}(i^{t+1}, j^{t+1}) + D_{m,n}^{\text{warp}}(i^{t+1}, j^{t+1})},$$  

(9)

where $D_{m,n}^{\text{pro}}$ is obtained from formula (2). Due to the motion of the camera, there is systematic depth inconsistency between the depth $D^t$ and $D^{t+1}$, so the projected depth $D_{m,n}^{\text{pro}}$ by the motion estimation $T_m$ is used to mend this inconsistency. $D_{m,n}^{\text{warp}}$ is obtained by bilinear interpolation like $X_{m,n}^{\text{warp}}$ in formula (3), in order to find the corresponding depth in depth map $D^t$ for $D^{t+1}$. Then, the depth inconsistency map is used to generate the occlusion weight mask:

$$M_{m,n}^{\text{weight}}(i^{t+1}, j^{t+1}) = 1 - D_{m,n}^{\text{diff}}(i^{t+1}, j^{t+1}).$$  

(10)

The losses of the inconsistent pixels will be assigned low weights by the occlusion weight mask. The binary auto-mask [14] is also used to filter the objects which are static relative to the camera and textureless regions, because these relatively static objects have different motion patterns with the overall pose estimation, and textureless regions make the networks confused about the pixel match for these regions. The binary auto-mask is calculated as follows:

$$M_{m,n}^{\text{auto}} = [\sigma(X_{m,n}^{\text{warp}}, X^{t+1}) < \sigma(X^t, X^{t+1})],$$  

(11)

where $[\cdot]$ represents the Iverson bracket. These static and textureless pixels are removed by removing the pixels with higher photometric loss between $X_{m,n}^{\text{warp}}$ and $X^{t+1}$ than the photometric loss between $X^t$ and $X^{t+1}$. The final masked image reconstruction loss of $D_n$ and $T_m$ is:

$$L_{m,n}^R = \frac{1}{|\Omega|} \sum_{\Omega} M_{m,n}^{\text{weight}} M_{m,n}^{\text{auto}} \sigma(X_{m,n}^{\text{warp}}, X^{t+1}),$$  

(12)

where $\Omega$ means all pixels for an image and $|\Omega|$ means the amount of pixels in an image. Depth consistency loss [13] is used to learn the consistent scale for the depth and pose in consecutive frames:

$$L_{m,n}^{\text{GC}} = \frac{1}{|\Omega|} \sum_{\Omega} M_{m,n}^{\text{auto}} D_{m,n}^{\text{diff}}.$$

(13)

Fig. 3. The details of forward warping. Given a random pose $T_{aug}$, pixels in the original images $X^t$ are projected to new pixel positions on the augmented images $X_{aug}^t$. The figure shows two notable issues in forward warping, i.e. holes and collisions.

$D_{m,n}^{\text{diff}}$ is obtained from formula (9), which measures the difference of depth estimation between two adjacent frames. Therefore, formula (13) encourage the depth consistency in two adjacent frames. The edge-aware depth smoothness loss is used to make the estimated depth smoother inside an object but remains unsmooth at the edge of the object. The revised depth smooth loss [20] is used to mitigate depth degradation in the training process:

$$L_{m,n}^{\text{smooth}} = \frac{1}{\Omega} \sum_{\Omega} \left\| \nabla \left( \frac{D^{t+1} - \text{min}(D^{t+1})}{\text{min}(\text{min}(D^{t+1}))} \right) e^{-\nabla X^{t+1}} \right\|.$$  

(14)

3) Multi-Scales Depths and Multiple Poses Association in Loss Functions: As the last refined pose $T_M$ is most accurate in the estimated multi-layer poses, $T_M$ is used to optimize all depths. However, coarser poses can have a negative effect on depth estimation in joint training. Therefore, stopping gradient is made for all scale depths, $D_1, D_2, \ldots, D_N$, when they are optimized with the coarser pose, $T_1, T_2, \ldots, T_{M-1}$. The overall image reconstruction loss and geometry consistency loss are:

$$L^R = \text{stop}_D \left( \sum_{u \in [1,2,\ldots,M-1]} L_{u,v}^R + \sum_{v \in [1,2,\ldots,N]} L_{M,v}^R \right),$$  

(15)

$$L^{GC} = \text{stop}_D \left( \sum_{u \in [1,2,\ldots,M-1]} L_{u,v}^{GC} + \sum_{v \in [1,2,\ldots,N]} L_{M,v}^{GC} \right),$$  

(16)

where $\text{stop}_D$ means stopping gradient for the input depth. The ablation study in Sec. IV-C will demonstrate the effectiveness of this design. For the convenience of explanation, only losses for $t+1$ are presented above. By projection in the opposite direction, losses for $t$ are also used.

C. 6-DoF Pose Augmentation by Means of Forward Warping

As the pose distribution of the KITTI dataset [22] is not uniform, a virtual dataset that includes an extensive distribution of pose data is proposed by Wang et al. [23]. However, the data gap is inevitable from the virtual data to the real autonomous driving scenario. Therefore, we propose a 6-DoF pose augmentation by generating a random pose $T_{aug}$ and then
The origin image $X^t$ to synthesizing a new arbitrary view image $X^t_{aug}$:

$$D^aug_{pro}(i_{pro}, j_{pro}) \cdot [i_{pro}, j_{pro}, 1]$$

$$= KT_{aug}(D'(i', j') \cdot K^{-1}[i', j', 1]^T),$$

(17)

where $K$ is the intrinsic matrix, $(i', j')$, $(i_{pro}, j_{pro})$ are respectively the pixel coordinates in the image plane of $X^t$ and $X^t_{aug}$. $D'(i', j')$ and $D^aug_{pro}(i_{pro}, j_{pro})$ are respectively the depth of $(i', j')$ and $(i_{pro}, j_{pro})$. The details of forward warping is shown in Fig. 3.

Due to occlusion in the transformation, collisions may occur, that is, more than one pixel in the original images may be projected to the same grid in the augmented images. To solve the collision problem, pixels with minimum depth in original images $X'$ are selected to be displayed on the augmented image $X^t_{aug}$. Apart from collisions, the occurrence of holes is another notable issue in forward warping, which means that pixels of the augmented image $X^t_{aug}$ may have no projection points from the original image $X'$. To handle holes, a novel hole filling method is established.

6-DoF pose augmentation increases the number of holes. In order to fill these holes, we first get a binary mask $H$, where pixels with holes in augmented images are labeled as 0 and others as 1 in the forward warping process. Then, to address the issue of background pixels filling the foreground image, which is mentioned in the work of Aleotti et al. [45], we convert $H$ to the binary mask $H'$. In $H'$, the black pixels are needed to be inpainted. Next, an inpainting strategy [58] is utilized to inpaint the image by applying the binary mask $H'$. However, we find that there is a large area of continuous holes at the edge of the augmented image as shown in Fig. 4(d), leading the inpainting strategy not accurate enough. Thus, we propose a novel hole filling method to handle this problem, the main idea of which is to fill the small area of holes by employing inpainting strategy and fill the large continuous holes at the edges of the figure with zero. Specifically, we first dilate mask $H'$ to $H_3$, and then get the binary inpainting mask $H_3$ by following formula:

$$H_3 = H_2 - H'.$$

Finally, we apply the inpainting strategy only to pixels labeled with 1 in $H_3$. An example of our hole filling method is shown in Fig. 4. Since Fig. 4(b) is the raw image to be inpainted and the binary mask $H_1$ labels pixels that needed to be inpainted, we multiply $H_3$ and Fig. 4(b) to get Fig. 4(g). In Fig. 4(g), the black cracked areas in the image are needed to be inpainted. Compared with Fig. 4(b), the black cracked areas contain areas that are not holes in Fig. 4(b). To address the issue of background pixels filling the foreground areas in the synthesized image [45], it is necessary to use Fig. 4(g) rather than Fig. 4(b) to apply the image inpainting method.

The augmentation pose is random but known between the original image and the augmented image. This property is used to train the PoseNets by supervised training. The original image and the augmented image are input to the PoseNets as supervised loss is used for this training.

$$L^pose_{aug} = ||t_m - t_{aug}||_2 \exp(-w_l) + w_l + ||q_m - q_{aug}||_2 \exp(-w_q) + w_q,$$

(19)

where $\cdot\cdot\cdot_2$ denotes the $l_2$-norm, $t_{aug}$ and $q_{aug}$ are respectively translation vector and quaternion generated by known random pose $T_{aug}$, $t_m$ and $q_m$ are respectively translation vector and quaternion generated by the pose estimation during the training process.
TABLE I

DEPTH EVALUATION RESULTS USING EIGEN ET AL. TEST SPLIT [8] ON KITTI DATASET. PREVIOUS SUPERVISED AND UNSUPERVISED MONOCULAR DEPTH LEARNING METHODS ARE COMPARED WITH OURS. D: DEPTH SUPERVISION. S: SELF-SUPERVISED FROM STEREO VIDEO. M: SELF-SUPERVISED FROM MONOCULAR VIDEO. AT: AUXILIARY TASK

| Methods                  | with AT | Train | AbsRel | SqRel | RMSE | RMSE log | \( \delta < 1.25 \) | \( \delta < 1.25^2 \) | \( \delta < 1.25^2 \) |
|--------------------------|---------|-------|--------|-------|------|----------|-----------------|-----------------|-----------------|
| Eigen et al. [8]         | M       | D     | 0.203  | 1.548 | 6.307| 0.282    | 0.702           | 0.890           | 0.958           |
| Liu et al. [26]          | M       | D     | 0.202  | 1.614 | 6.523| 0.275    | 0.678           | 0.895           | 0.965           |
| Garg et al. [29]         | M       | S     | 0.152  | 1.226 | 5.849| 0.246    | 0.784           | 0.921           | 0.967           |
| Godard et al. [30]       | M       | S     | 0.146  | 1.344 | 5.927| 0.247    | 0.803           | 0.922           | 0.964           |
| Zhan et al. [31]         | M       | S     | 0.144  | 1.391 | 5.869| 0.241    | 0.803           | 0.928           | 0.969           |
| Kuznietsov et al. [27]   | M       | DS    | 0.113  | 0.741 | 4.621| 0.189    | 0.862           | 0.960           | 0.986           |
| Guo et al. [28]          | M       | DS    | 0.096  | 0.641 | 4.095| 0.168    | 0.892           | 0.967           | 0.986           |
| Tian et al. [2]          | M       | S     | 0.095  | 0.601 | 4.128| 0.176    | 0.908           | 0.976           | 0.991           |

quat erion generated by the pose \( T_{m} \). \( \omega_{t} \) and \( \omega_{q} \) are learnable parameters for translation vector and quaternion respectively.

IV. EXPERIMENTS

A. Datasets and Implementation Details

1) KITTI Raw Dataset: The KITTI raw dataset [22] contains raw color images and depth sensor information collected, and is divided into the categories ‘Road’, ‘City’, ‘Residential’, ‘Campus’, and ‘Person’. For the evaluation of depth estimation, we follow previous work [13] to train and test models on the data split of Eigen et al. [8] only on KITTI raw dataset [22].

2) KITTI Odometry Dataset: The KITTI odometry splits [22] contains 11 long driving stereo sequences with available ground truth trajectories. For pose estimation, we evaluate visual odometry results on KITTI odometry dataset [22] as previous works [13], [62], using sequences 00-08 for training and sequences 09-10 for testing. Following the KITTI odometry dataset evaluation criterion [63], we report the average translational and rotational errors on possible sub-sequences of trajectory length of 100, 200, ..., 800 meters.

3) Implementation Details: All the experiments for 2 PoseNets are implemented on a NVIDIA RTX 2080Ti GPU. All the experiments about 3, 4, 5, and 6 PoseNets are implemented on a NVIDIA RTX 3090 GPU. For pose estimation, separate convolution neural networks in [13] are used to estimate coarse pose and residual poses between two adjacent images. For depth estimation, DispResNet18 architecture [13] is used to generate multi-scale disparity maps from a single image. The image size for all experiments is 832 × 256. A snippet of two sequential images is used to estimate both forward pose and backward pose between two frames. The losses are used in two-directional transformation and image reconstruction like SC-SFM [13]. The hyperparameters are set as \( \alpha_1 = 1.0 \), \( \alpha_2 = 0.1 \), \( \alpha_3 = 0.5 \), and \( \alpha_4 = 2.0 \) in formula (1), \( \lambda_{\rho} = 0.15 \) in formula (8). The initial value of the learnable parameters \( \omega_{t} \) and \( \omega_{q} \) are set as 0.0 and \( -2.5 \) respectively. Adam optimizer [64] is used for training, and the learning rate is set as \( 10^{-4} \). The batch size is 4 for all experiments.

B. Evaluation Results

Through the training and test as in Sec. IV-A, our depth estimation results and visual odometry results on the KITTI dataset are shown in Tables I and II, respectively. On both monocular depth estimation and monocular visual odometry by the unsupervised learning of depth and pose, our method outperforms recent state-of-the-art methods. Besides, our method does not need auxiliary tasks. On the contrary,
many methods like [17], [18], [19], [20], [35], and [59] require additional optical flow network to perform geometric consistency between optical flow, depth, and pose. ΩNet [59] requires manually annotated semantic segmentation of ground truth. CC [19] and EPC++ [35] require additional dynamic mask network. Our method uses only monocular images for full self-supervised learning of depth and pose, and achieves better results. We explicitly enhance the pose estimation accuracy by iterative pose refinement with view synthesis and pose augmentation, which leads to better depth estimation performance through joint learning losses. We give the qualitative results of our depth estimation compared with other methods, as shown in Fig. 6. Our methods perform better in many regions such as cars, roads, guideboards, buildings, and the
TABLE III
ABLATION RESULTS FOR DEPTH ESTIMATION USING EIGEN ET AL. TEST SPLIT [8] ON KITTI DATASET. A-DEPTH: ALL-SCALE DEPTH, A-POSE: ALL POSES, H-POSE: HIGHEST-REFINED POSE, C-POSE: COARSE POSES, WHICH ARE THE POSES EXCEPT FOR THE HIGHEST-REFINED POSE. ⇆: LOSSES AND GRADIENT ARE CALCULATED WITH TWO SIDES, →: LOSSES ARE CALCULATED WITH TWO SIDES, BUT GRADIENTS ARE ONLY CALCULATED FOR THE RIGHT SIDE AND STOPPED FOR THE LEFT SIDE.

| Methods                                      | AbsRel | SqRel | RMSE | RMSE Log | \(\delta < 1.25\) | \(\delta < 1.25^4\) | \(\delta < 1.25^\infty\) |
|----------------------------------------------|--------|-------|------|----------|---------------------|----------------------|------------------------|
| Baseline (Single PoseNet)                    | 0.116  | 0.848 | 4.919| 0.195    | 0.868               | 0.957                | 0.981                   |
| (a) A-depth \(\Leftarrow\) A-pose            | 0.113  | 0.815 | 4.786| 0.190    | 0.973               | 0.959                | 0.982                   |
| A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose | 0.111  | 0.813 | 4.656| 0.189    | 0.879               | 0.960                | 0.982                   |
| (b) 2-pose (with A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.111  | 0.813 | 4.656| 0.189    | 0.879               | 0.960                | 0.982                   |
| 2-pose + Pose augmentation (with A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.110  | 0.810 | 4.709| 0.187    | 0.881               | 0.962                | 0.982                   |
| (c) 2-pose (with pose augmentation + A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.110  | 0.810 | 4.709| 0.187    | 0.881               | 0.962                | 0.982                   |
| 3-pose (with pose augmentation + A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.109  | 0.834 | 4.692| 0.186    | 0.883               | 0.962                | 0.982                   |
| 4-pose (with pose augmentation + A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.109  | 0.790 | 4.656| 0.185    | 0.882               | 0.962                | 0.983                   |
| 5-pose (with pose augmentation + A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.110  | 0.859 | 4.700| 0.188    | 0.884               | 0.981                | 0.983                   |
| 6-pose (with pose augmentation + A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose) | 0.109  | 0.800 | 4.765| 0.188    | 0.880               | 0.960                | 0.982                   |

TABLE IV
TEST TIME, NUMBER OF PARAMETERS, AND FLOPS COMPARISON WITH SC-SFM [13] AND MONODEPTH2 [14]. THE TESTING TIME IS AVERAGED ON 500 ITERATIONS WITH TEST BATCH SIZE AS 1.

| Methods                | SC-SFM [13]               | Monodepth2 [14]             | Ours                      |
|------------------------|---------------------------|----------------------------|---------------------------|
| Network                | DepthNet                  | PoseNet                    | DepthNet                  | 2-PoseNets | 3-PoseNets | 4-PoseNets | 5-PoseNets | 6-PoseNets |
| Test Time              | 5.5ms                     | 3.4ms                      | 5.5ms                     | 5.5ms      | 8.0ms      | 12.7ms     | 17.2ms     | 21.3ms     | 26.1ms     |
| Test Frequency         | 192Hz                     | 292Hz                      | 180Hz                     | 284Hz      | 192Hz      | 124Hz      | 78Hz       | 58Hz       | 460Hz      | 38Hz       |
| Parameters Number      | 14.3M                     | 12.5M                      | 14.3M                     | 12.5M      | 14.3M      | 25.0M      | 37.5M      | 30.0M      | 62.5M      | 75.0M      |
| Flops                  | 13.9G                     | 8.5G                       | 21.4G                     | 8.5G       | 13.9G      | 17.0G      | 25.5G      | 34.0G      | 42.5G      | 51.0G      |

sky. The qualitative results of our visual odometry are shown in Fig. 7.

C. Ablation Study

All ablation studies use the same method to train and test DepthNet as Sec. IV-B.

1) Multi-Scale Depth and Multiple Pose Association: As there are many ways to associate multi-scale depth maps with several refined poses, different types of association between poses and depth maps are compared based on 2 PoseNets in the ablation study as shown in Tables III(a). (1) A-depth \(\Leftarrow\) A-pose: All depths and all poses are used for losses, and the gradient is calculated for all. (2) A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose: All poses are used for losses with all depths, but the gradient is stopped for all depths when associating with coarse poses. The results in Tables III(a) show that “A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose” has the best results because this method solves the problem that coarse poses may bring in disturbance to the depth estimation as described in Sec. III-B.

2) Pose Augmentation: We test the effects of our augmentation methods in the case of 2 PoseNets “A-depth \(\Leftarrow\) H-pose, A-depth \(\rightarrow\) C-pose”. The results in Tables III(b) show that pose augmentation learning brings better performance as the data augmentation brings a wider data distribution of the training set. Therefore, this pose augmentation method is as the baseline for the following experiments.

3) Number of Pose Refinement: We add more residual PoseNets for more pose refinements and test the results, including 2, 3, 4, 5, and 6 PoseNets, as shown in Tables III(c). These PoseNets all have the same network structure as the PoseNet in Monodepth2 [13]. The results show that more pose refinements benefit the results for 1-4 PoseNets. However, the gain of the number of networks on the results is not obvious when the number of PoseNets increases higher than 4. Since the pose estimation obtained by 4-PoseNets is already accurate enough, the generated intermediate view images \(x_{m,n}^{\text{warp}}\) are very close to the original images \(x_{t+1}\) at time \(t+1\). Limited to the capacity of PoseNet, using one more PoseNet to estimate the residual pose instead introduces a more significant error in the final results.

D. Efficiency

The DepthNet gets the best performance with 4-PoseNets considering all the evaluation metrics according to the ablation experiments. For better performance, the fixed 4-PoseNets are adopted compared with other methods. As we use the same DepthNet architecture as [13], the multiple-scales DepthNet will not affect system efficiency. The multiple-PoseNets reduce the efficiency of pose estimation. But importantly, the test efficiency of depth estimation will not be influenced because DepthNet and multiple-PoseNets are used separately during testing, and both our DepthNet and PoseNets meet the real-time requirements for a 20Hz camera (The camera of KITTI dataset [22] is 20Hz). We evaluate the test time on NVIDIA Geforce RTX 3090 GPU. The comparison results of test time, number of parameters, and flops with SC-SFM [13] and Monodepth2 [14] are listed in Table IV.

V. CONCLUSION

The mutual coupling learning based on multi-depth and multi-pose losses is proposed in this paper to refine pose layer by layer. The synthesized middle view narrows the distance between adjacent frames of images, making residual pose estimation easier, thereby improving the effect of pose estimation. Because of the coupling between depth and pose estimation, depth estimation has also been further improved. In addition, by synthesizing intermediate view images, 6-DoF pose data augmentation is proposed for this unsupervised
learning framework, which enlarges the distribution of pose data. To our knowledge, this paper is the first to realize augmenting the 6-DoF pose in pose learning on 2D images. Finally, our framework achieves state-of-the-art performance on KITTI dataset without any other auxiliary task.

REFERENCES

[1] S. Chen, Z. Pu, X. Fan, and B. Zou, “Fixing defect of photometric loss for self-supervised monocular depth estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 3, pp. 1328–1338, Mar. 2022.

[2] F. Tian et al., “Depth estimation using a self-supervised network based on cross-layer feature fusion and the quadtree constraint.” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 4, pp. 1751–1766, Apr. 2022.

[3] G. Wang, X. Wu, Z. Liu, and H. Wang, “PWCL-Net: Deep LiDAR odometry in 3D point clouds using hierarchical embedding mask optimization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15910–15919.

[4] M. Xu, A. Hu, and H. Wang, “Image-based visual impedance force control for contact aerial manipulation,” IEEE Trans. Autom. Sci. Eng., early access, Apr. 6, 2022, doi: 10.1109/TASE.2022.3162207.

[5] G. Wang, Y. Yang, H. Zhang, Z. Liu, and H. Wang, “Spherical interpolated convolutional network with distance-features density for 3-D semantic segmentation of point clouds,” IEEE Trans. Cybern., early access, Nov. 8, 2021, doi: 10.1109/TCYB.2021.3124954.

[6] Z. Liu, H. Wei, H. Wang, H. Li, and H. Wang, “Integrated task allocation and path coordination for large-scale robot networks with uncertainties,” IEEE Trans. Autom. Sci. Eng., vol. 19, no. 4, pp. 2750–2761, Oct. 2022.

[7] K. Li, H. Wang, X. Liang, and Y. Miao, “Visual servoing of flexible-link manipulators by considering vibration suppression without deformation measurements,” IEEE Trans. Cybern., vol. 52, no. 11, pp. 12454–12463, Nov. 2022.

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

[9] M. Song, S. Lim, and W. Kim, “Monocular depth estimation using Laplacian pyramid-based depth residuals,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 11, pp. 4381–4393, Nov. 2021.

[10] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “UNet: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Assist. Interv., 2017, pp. 424–432.

[11] Y. Almalioglu, M. R. U. Saputra, P. P. B. D. Guimiao, A. Markham, and Y. Almalioglu, “GANVO: Unsupervised deep monocular visual odometry,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 8978–8986.

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

[13] Y. Zou, Z. Luo, and J.-B. Huang, “DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 36–53.

[14] G. Wang, C. Zhang, H. Wang, J. Wang, Y. Wang, and X. Wang, “Unsupervised learning of depth, optical flow and pose with occlusion from 3D geometry,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 1, pp. 308–320, Jan. 2022.
