2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

D^{2}VO: Monocular Deep Direct Visual Odometry

Qizeng Jia[1*], Yuechuan Pu[1*], Jingyu Chen[1], Junda Cheng[1], Chunyuan Liao[2], Xin Yang[1†]

Abstract— **In this paper, we present a novel deep learning and direct method based monocular visual odometry system named D^{2}VO. Our system reconstructs the dense depth map of each keyframe and tracks camera poses based on these keyframes. Combining direct method and deep learning, both tracking and mapping of the system could benefit from the geometric measurement and semantic information. For each input frame, a feature pyramid is built and shared by both tracking and mapping process. The depth map of keyframe is efficiently estimated from coarse to fine with the followed multi-view hierarchical depth estimation network. We optimize the camera pose by minimizing photometric error between re-projected features of each frame and its reference keyframe with bundle adjustment. Experimental results on TUM dataset demonstrate that our approach outperforms the state-of-the-art methods on both tracking and mapping.**

I. INTRODUCTION

Visual odometry (VO) enables robots to perceive the surrounded environment and determine its localization with a light-weighted camera, thus is essential for robotics, automatic driving and autonomous flight of UAVs. In the last couple of years, several classic VO systems have been proposed and have proven their good performance under a large number of scenarios. Simultaneous localization and mapping system (SLAM) can be regarded as an extended version of VO, which implements the same function but applies additional modules such as loop closing. Both of these systems are built based on two fundamental interdependent processes, i.e. tracking and mapping. The mapping process reconstructs the structure of environment and the tracking process calculates the camera pose based on the known depth structure. According to the different tracking and mapping methods, VO can be divided into two categories, i.e. feature based method and direct method. Feature based methods, e.g. ORB-SLAM [1], find corresponding feature points and track camera pose via solving PnP problem. Their mapping process calculate the depth of those feature points with triangulation. Direct method based VO, e.g. LSD-SLAM [2], does not need to extract feature points, the system obtains depths on high-textured region with epipolar line search by minimizing the photometric error between current frame and its reference keyframe. Both tracking and mapping of these methods are based on geometric calculation.

With rapid development of deep learning, deep neural network has exhibited its strong ability in plenty of computer vision tasks, e.g. classification, recognition, semantic segmentation, stereo vision, etc. With those advancements, plenty of learning based methods have been proposed to solve dense mapping and camera tracking. Single view depth estimation networks [3], [4], [5], [6], [7] direct infer the depth map from a single input RGB image. Different with conventional geometric methods, these methods attempt to learn a mapping relationship between the RGB image and its depth map. The networks are trained with mass of data and regress the depth value of each pixel of the input RGB image. [8], [9] investigate the potential of combining stereopsis cues and the learned structure priors from a single-view depth CNN. These approaches combine multi-view stereo cues and single-view priors in a loosely coupled manner. Multi-view depth estimation networks [10], [11], [12] aim to estimate depth with known camera poses. The geometric information is embeded into the network by inputting reference image along with its cost volume that are calculated with a series of frames with known poses. To estimate the ego-motion of camera, deep learning based methods always utilize a pose network with two frames as input and output the relative pose between them. Single view depth estimation network and pose network can easily form up a tracking and mapping system similar to VO, e.g. [13], [14], [15], [16]. In contrast to traditional VO system, without geometric cues, the depth prediction is purely based on priori knowledge of training dataset. Also, the pose network is too ambitious to predict camera motion with only two RGB images. Another problem is that both networks do not encode camera intrinsic information, which means the system cannot be generalized to other camera with different intrinsic parameters. To solve those limitations, [17] proposed to replace the pose net with direct minimizing photometric error to obtain camera pose. BA-Net [18] embeded bundle adjustment(BA) into a single view depth estimation network. Rather than using a pose net, direct method which estimate camera pose with BA are more practical and reliable.

With geometric calculation, state-of-the-art traditional VO systems are robust and performs well for camera tracking in most scenarios. However, the reconstructed depth structure is incomplete, since only depths of feature points or in high-textured region are measured. CNN-based depth and pose prediction methods could output dense depth map and camera pose, but the pose or depth prediction may be purely based on semantic information learned from training

This work was supported by the National Natural Science Foundation of China (61872417), the Fundamental Research Funds for the Central Universities (2019kfyRCPY118, 2020kfyXGYJ026), the Open Project of Wuhan National Laboratory for Optoelectronics (2018WNLOKF025).

† Corresponding author: Xin Yang (xinyang2014@hust.edu.cn)

*Authors contribute equally

[1] Qizeng Jia, Yuechuan Pu, Jingyu Chen, Junda Cheng and Xin Yang are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei, China, 430074.

[2] Chunyuan Liao is with HiScene Information Technology,Co., Ltd, Shanghai, China.

dataset. Without geometric measurement, the depth or pose prediction may crash in unfamiliar or unseen scenarios. Toward practical CNN-based VO, we aim to encode geometric information in both tracking and mapping process with deep learning. We combine the advantages of both deep learning and the traditional method in our VO system. To this end, we propose deep direct visual odometry, denoted as D^2VO. For dense mapping, we develop a multi-view depth estimation network. The network takes frames with corresponding camera poses as input. The multi-view geometric information is encoded with the calculated cost volume based on camera poses. To maintain both efficiency and accuracy, the depth map is estimated hierarchically from coarse to fine. With the RGB frame and its estimated dense depth, the following camera pose can be optimized with bundle adjustment(BA) by minimizing the photometric error between current frame and re-projected reference frame. To utilize the advantage of CNN, the photometric error is calculated between features of two frames. The features are extracted with a neural network and shared by depth estimation network for efficiency. Based on this multi-view depth estimation network and the feature based direct method, we design the pipeline of our keyframe based VO as follows. In our system, we only estimate the depth of keyframes. The depth of this keyframe is estimated with its previous frames, and the following frames are tracked with direct method based on this keyframe. The tracking and mapping process are coupled as the traditional method, and our system gets semantic information with CNN and could output dense depth map. For system initialization, we design an initialization method with the same depth network used for mapping. Without additional network for initialization, the storage space of the program can be saved.

To summarize, our D^2VO uses an efficient multi-view depth estimation network for mapping and feature map based direct method for tracking. Both depth and pose are estimated based on geometric measurement in our system, thus our system combines the advantage of CNN and traditional methods. We also design an initialization method with the same depth estimation network. The experimental results demonstrate our system outperforms the state-of-the-art methods.

II. RELATED WORK

In this section, we investigate related works to our D^2VO, e.g. traditional VO/SLAM systems, CNN-based depth and pose prediction methods and learning-based VO systems.

State-of-the-art VO/vSLAM systems can be categorized into two classes: feature-based method and direct method. Notable feature-based methods include PTAM [19] and ORB-SLAM [1]. These methods estimate camera pose by detecting sparse feature points and finding correspondences between current frame and local map and applying PnP algorithm to the feature correspondences. With estimated camera poses, the depth of each feature point could be calculated via triangulation. Direct methods such as LSD-SLAM [2] and DSO [20] find correspondences on high-textured region, resulting in a much denser depth map. Both methods optimize camera trajectory with BA to minimize photometric error. To obtain dense depth map, DTAM [21] features a standard multi-view dense depth/disparity estimation pipeline including cost volume computation, cost volume aggregation, depth estimation and depth refinement. Visual-inertial systems (VINs) [22] is a method based on low-cost IMUs which can provide accurate camera motions in real-time. [23] presents an IMU pre-integration correction approach which reduces the negative impact of IMU noises.

The great progress of deep learning stimulates the learning-based depth estimation methods. Eigen et al. [3] was the first to employ a two-stage network architecture for single-view depth prediction. This two-stage network was further improved in [4] to output both depth normal and the depth map. Laina et al. [5] improved the accuracy of single-view depth estimation with a deep CNN based on ResNet. DORN [6] proposed a spacing-increasing discretization strategy to discretize depth, recasted depth network learning as an ordinal regression problem. To alleviate the difficulties in collecting data with depth ground truth, Garg et al. [7] developed an unsupervised training method for depth estimation using a objective function to minimize the photometric error between stereo image pairs.

Although CNN-based depth prediction network could infer dense depth map for every pixel, the accuracy and generalization ability of these methods are still worrisome. The network learns depth of single image with only priori knowledge in training datasets, resulting in poor performance in unseen scenarios. To alleviate this problem and combine the advantage of geometric and semantic information, some approaches combined single-view depth estimation with mature traditional SLAM/VO system. Yang et al. proposed a Bayesian DeNet [9] which computed depth and the corresponding uncertainty using a single-view depth CNN, multi-view depth measurements were then fused in a Bayesian framework. CNN-SLAM [24] and CNN-SVO [25] used depth map from single-view depth estimation as an initialization of depth estimation. [26], [27] implemented dense mapping by inputting RGB image with sparse/semi-dense depth map obtained from ORB-SLAM/LSD-SLAM to depth estimation network respectively.

To better integrate geometric cues into CNN, multi-view depth estimation networks predict depth with multiple input frames with known camera poses. DeepMVS [28] divided an input RGB image into small image patches and input the reference patches and their candidate matching patches on the epipolar line of neighbor images to the network for finding matching correspondences. MVDepthNet [10] computed the cost volume of a reference image using the conventional plane sweep algorithm and then input the reference image along with its cost volume into a lightweight CNN for accurate depth estimation. MVSNet [11] calculated cost volume on feature map with differentiable re-projection layer. The cost volume was then fed into a CNN with 3D convolution. DeMoN [29] presented a depth and motion network which explored the stereo cues by alternating optical flow estimation with the estimation of camera motion and depth. The optical net found dense pixel correspondences

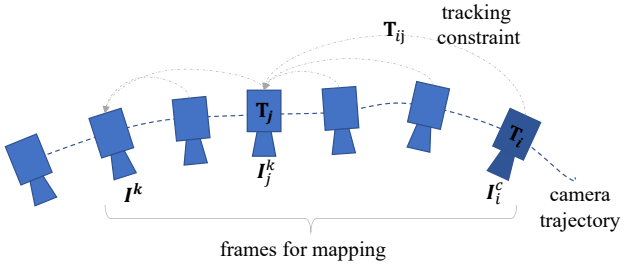

Fig. 1. Illustration of tracking and mapping of frame $\mathbf{I}_i^c$, we track relative pose $\mathbf{T}_{ij}$ between $\mathbf{I}_i^c$ and its keyframe $\mathbf{I}_j^k$. Once $\mathbf{I}_i^c$ is determined as a new keyframe, we estimate the depth of $\mathbf{I}_i^c$ with all the frames between $\mathbf{I}_i^c$ and $\mathbf{I}_k$.

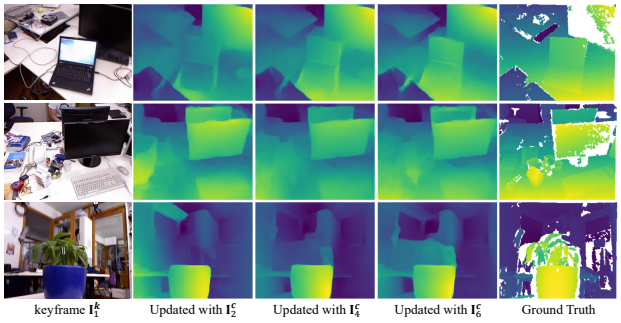

Fig. 2. Three examples of estimated depth map of the keyframe $\mathbf{I}_1^k$ during initialization. As the initialization progresses, the depth map of the initial frame is continuously improved.

and depth map was calculated with triangulation and refined with depth net. DENAO [30] tightly coupled a single-view depth net with an optical flow net as an auxiliary helper.

Some methods also proposed to estimate camera pose based on CNN. DeepVO [31] employed an RNN to predict camera pose end-to-end from input image sequence. Many methods, e.g. Tinghui et al. [13], GeoNet [14], sfm-net [15], UndeepVO [16], concurrently trained a pose net and a depth net to form CNN-based VO systems. The depth net and pose net were jointly trained and provide training constraint to each other in an unsupervised manner. The depth network took a single image as input to predict depth map and the pose network took two successive images as input to predict the relative pose between them. However, the depth net and pose net were performed separately during deployment. Without geometric cues, predicting pose directly from two images is difficult. To avoid the pose prediction difficulties, Wang et al. [17] replaced the pose prediction network with direct method, where the camera pose was calculated with BA. BA-Net [18] proposed to integrate the BA option into a deep neural network. They predicted the depth of image with single-view depth estimation method but optimized the camera pose with BA to minimize photometric error between the features of image pairs. To force the network to learn multi-view geometric information, DeMoN [29] predicted the camera pose from a branch of optical flow net. DeepTAM [12] iteratively updated poses with predicted pose residuals between reference frame and warped neighbor frame from a pose net.

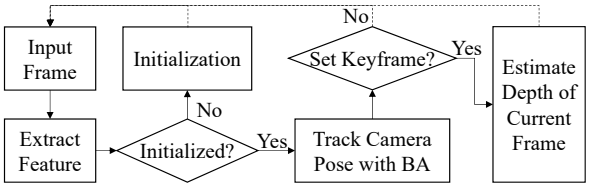

Fig. 3. Illustration of system pipeline. For each input frame, the system first extracts its feature map. When the system is not initialized, it enters the initialization process. If the system has been initialized, the system tracks the camera pose and predict the depth of the key frames.

III. METHOD

In this section, we illustrate the proposed D^{2}VO in detail. We first introduce the system pipeline of initialization, tracking and mapping. Then we describe the network architecture for dense multi-view depth estimation in mapping process and the direct method based tracking procedure.

A. System Pipeline

As illustrated in Fig. 1, for i^{th} coming camera frame $\mathbf{I}_i^c$, we aim to estimate its corresponding camera pose $\mathbf{T}_i = [\mathbf{R}_i, \mathbf{t}_i] \in \mathbb{SE}(3)$ in world coordinate, composed of a 3×3 rotation matrix $\mathbf{R}_i \in \mathbb{SO}(3)$ and a 3D translation vector $\mathbf{t}_i \in \mathbb{R}^3$. The camera poses are tracked based on its corresponding keyframe $\mathbf{I}_j^k$, which is a frame before $\mathbf{I}_i^c$ with calculated camera pose $\mathbf{T}_j$ and inverse dense depth map $\mathbf{D}_j$. The inverse depth of $\mathbf{I}_j^k$ is estimated with a multi-view depth estimation network in mapping procedure previously. With camera trajectory and dense depth map $\mathbf{D}$ of each keyframe, the robot or other equipment is enabled to move autonomously in an unfamiliar environment.

For a keyframe based monocular VO/SLAM system, tracking, i.e. camera pose estimation, is based on the depth of reference keyframe or local map and mapping, i.e. depth estimation is based on the known camera poses of previous frames of the current keyframe. Thus, tracking and mapping procedure are necessary to each other. Without neither known camera pose or depth map, the first important thing for a VO/SLAM is to initialize the system with an initial depth map and camera pose. In ORB-SLAM [1], the initial camera pose is calculated with solving homograph or essential matrix. DeepTAM [12] focuses on solving both tracking and mapping with CNN, for initialization, they uses an independent single-view depth estimation network. This method needs an additional large depth network and cannot avoid the inherent problem of single-view depth estimation.

We aim to design a CNN-based initialization method which integrates multi-view geometric cues but without redundant network. To this end, as same as LSD-SLAM [2], we randomly initialize the depth map $\mathbf{D}_1$ of the first input frame $\mathbf{I}_1^c$. The depth value $d_{u,v}$ of the pixel on $[u, v]^\top$ in this initialized depth map obeys gaussian distribution, i.e. $d_{u,v} \sim N(1, \sigma^2)$, where σ is set as 0.1 in our system. We then set frame $\mathbf{I}_1^c$ as a keyframe $\mathbf{I}_1^k$ and track the following input frame $\mathbf{I}_i^c$ respect to it. And with the determined camera

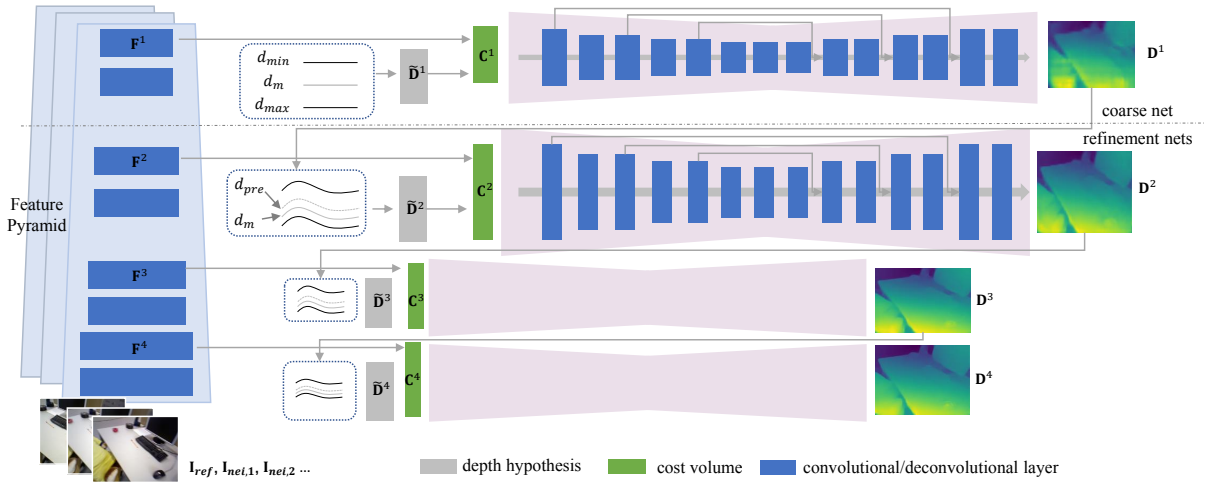

Fig. 4. The architecture of depth estimation network. The depth estimation network contains four sub-networks. The coarse net estimates the initial depth with the lowest feature resolution, and then three refinement nets gradually improve the depth prediction results of the previous layer.

pose $\mathbf{T}_i$ of $\mathbf{I}_i^c$, we input frame pairs (i.e. $\mathbf{I}_i^c$ and $\mathbf{I}_1^k$) and $\mathbf{T}_i$ into the mapping network and replace original depth map $\mathbf{D}_1$ with the network output. Then we track subsequent frames with updated depth map $\mathbf{D}_1$. With this iterative updating strategy(see Fig. 2), the depth map $\mathbf{D}_1$ is continuously meliorated based on the better camera pose and the pose of subsequent frames is also improved simultaneously with the improving reference depth map. For simplicity, we update the depth map $\mathbf{D}_1$ of the first keyframe $\mathbf{I}_i^k$ with fixed number of frames. When the pose of 10^{th} frame $\mathbf{I}_{10}^c$ is determined, we set $\mathbf{I}_{10}^c$ as the new keyframe $\mathbf{I}_{10}^k$. Then we estimate the depth map of this new keyframe by feeding features of $\mathbf{I}_{10}^k$, $\mathbf{I}_1^k$ and the relative pose $\mathbf{T}_{10,1}$ into the network.

After initialization, the following tracking is based on the reference keyframe. For input frame $\mathbf{I}_i^c$ to our system, we firstly utilize a neural network with 8 layers to extract the features. Every two layers, the resolution of the feautre map is reduced half with stride being set to 2. We take feature maps of each resolution to build a multi-scale feature pyramid $F_i^n(n = 1, 2, 3, 4)$. The camera pose is determined by minimizing the photometric residual of features between current frame and its reference keyframe. The detailed tracking process will be introduced in Sec. IIIC. Different to the initialization part, we do not update the depth of keyframe after obtaining the pose of a new frame for efficiency.

After tracking several frames, the distance between the new frame to its reference keyframe may get larger and overlapping rate between two frames decreases. At this time, the current keyframe is not suitable for further tracking, so we need to determine a new keyframe. To this end, a distance coefficient $\mathcal{D}$ is defined to measure the distance between the current frame $\mathbf{I}_i^c$ to its reference keyframe $\mathbf{I}_j^k$:

$$\mathcal{D} = \|\mathbf{R}_i^{c-1}\mathbf{t}_i^c - \mathbf{R}_j^{k-1}\mathbf{t}_j^k\|_2 \tag{1}$$

If $\mathcal{D} > 0.15$ or the rotation angle between $\mathbf{I}_i^c$ and $\mathbf{I}_j^k$ is larger than 6°, we determine the current frame $\mathbf{I}_i^c$ as a new keyframe $\mathbf{I}_i^k$.

For a newly determined keyframe, we need to calculate its depth map for the next tracking. We input all the features of frames between the current frame $\mathbf{I}_i^k$ and the second prior keyframe with their camera poses to our depth network. The network produces the depth map of the current keyframe end-to-end. The full system pipeline is illustrated in Fig. 3.

B. Mapping network architecture

We present the details of our multi-view depth estimation network architecture in this subsection. The depth network takes the feature map of reference keyframe $\mathbf{F}_{ref}^n$, the features $\mathbf{F}_i^n(i = 1, 2, ..., N)$ of N previous sequential neighbor frames along with their poses $\mathbf{T}_{ref}$ and $\mathbf{T}_i^n(i = 1, 2, ..., N)$ and camera intrinsic $\mathbf{K}$ as input. The depth map $\mathbf{D}_{ref}$ of $\mathbf{I}_{ref}^k$ is estimated following a hierarchical coarse-to-fine strategy. As shown in Fig. 4, we utilize four sub-networks to estimate depth. We call the sub-network based on the top of feature pyramid $\mathbf{F}^1$ with lowest feature resolution as coarse net, and denote other three sub-networks as refinement net. The coarse net estimates the initial depth and the other refinement nets refine depth result based on this initial depth map.

To encode multi-view geometric information into the depth network, we first construct the cost volume in each sub-network. To form the cost volume, we need to warp the neighbor feature to reference feature with respect to different depth hypothesis. The depth hypothesis $\widetilde{\mathbf{D}}_{ref}$ of reference keyframe is sampled in discrete interval. Without a prior depth for coarse depth net, the depth hypotheses are divided with a fixed depth range. The depth value d_m^n of a pixel in m^{th} depth hypothesis and n^{th} pyramid level could be calculated as:

$$d_m^n = d_{min} + m * \frac{d_{max} - d_{min}}{M - 1}, (m = 0, 1, ..., M - 1) \tag{2}$$

where d_{min} and d_{max} are the minimum and maximum depth value respectively. We take M depth hypotheses for each depth map.

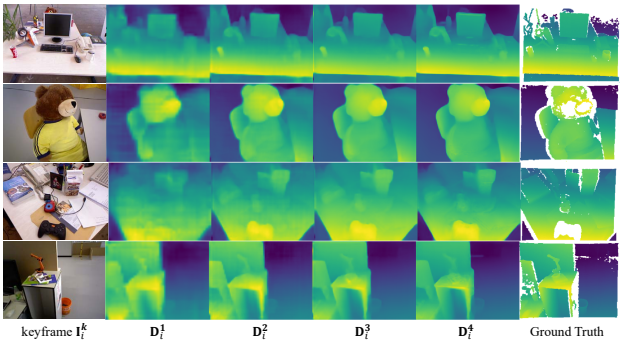

Fig. 5. The depth estimation results from sub-networks in different pyramid levels. $\mathbf{D}_i^1$ represents the depth prediction result output by the coarse net, then the depth maps in $\mathbf{D}_i^2$, $\mathbf{D}_i^3$, $\mathbf{D}_i^4$ are gradually refined.

For the refinement net, the search range of depth on the epipolar line could shrink to a narrower band based on the previously predicted depth. Thus, the depth hypothesis value d_m^n could be calculate as:

$$d_m^n = d_{pre} + (m - \frac{M}{2}) * \sigma * d_{pre}, (m = 0, 1, ..., M-1) \quad (3)$$

where d_{pre} is the previous depth of the last pyramid and σ is the depth sample interval.

Corresponding to m^{th} depth hypothesis $\widetilde{\mathbf{D}}_m^n$ of reference keyframe at n^{th} pyramid level, we warp each pixel $\mathbf{u}_{nei}^n$ on neighbor feature $\mathbf{F}_{nei,i}^n$ to the reference feature $\mathbf{F}_{ref}^n$ to form a new feature $\mathbf{F}_{nei,i}^{n,m}{}'$ using warping function:

$$\mathbf{u}_{nei}^{n,m'} = \pi(\mathbf{T}_{ref\sim nei,i} \cdot \pi^{-1}(\mathbf{u}_{nei}^n, \widetilde{\mathbf{D}}_m^n)) \quad (4)$$

where $\mathbf{T}_{ref\sim nei,i}$ is the relative camera pose between $\mathbf{I}_{ref}^k$ and $\mathbf{I}_{nei,i}^c$, π is the camera projection model which projects a 3D point in the world coordinate to a 2D coordinate on the image frame, π^{-1} is the inverse projection model that recovers 3D point in the camera coordinate from its 2D projection.

Assume the dimension of the feature map $\mathbf{F}_{nei,i}^n$ is $H \times W \times C$, the warped feature $\mathbf{F}_{nei,i}^{n,m}{}'$ with M depth hypotheses form up the feature map $\mathbf{F}_{nei,i}^n{}'$ in size of $H \times W \times C \times M$. The cost volume $\mathbf{C}$ is calculated based on $\mathbf{F}_{nei,i}^n{}'$ and $\mathbf{F}_{ref}^n$. Some methods directly build 4-dimensional cost volumes and perform 3D convolution. Instead, to maintain computational efficiency, we use the group wise average proposed in [32]. Specifically, we divide the feature maps of both reference frame and warped neighbor frames into G groups respect to the channel dimension. In each feature group with size of $H \times W \times C/G \times M$, the mean and variance of feature can be calculated as:

$$\mathbf{F}_{mean,g}^n = \frac{\mathbf{F}_{ref,g}^n + \sum_{i=1}^N \mathbf{F}_{nei,i,g}^{n,m}{}'}{N+1} \quad (5)$$

$$\mathbf{F}_{var,g}^n = \frac{|\mathbf{F}_{ref,g}^n - \mathbf{F}_{mean,g}^n|^2 + \sum_{i=1}^N |\mathbf{F}_{nei,i,g}^{n,m}{}' - \mathbf{F}_{mean,g}^n|^2}{N+1} \quad (6)$$

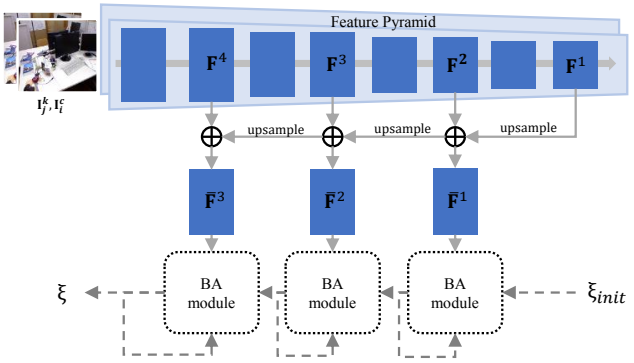

Fig. 6. Illustration of tracking procedure. Tracking procedure shares the same feature pyramid as the depth prediction network. The obtained feature $\overline{\mathbf{F}}^n (n = 1, 2, 3)$ will be input to the BA module for camera tracking. At each pyramid level, we iteratively update the camera pose then input the result to the next level for further refinement

We set the feature variance $\mathbf{F}_{var,g}^n$ as the cost volume $\mathbf{C}_g^n$ and average the value along its channel dimension. The size of averaged cost volume in each group is $H \times W \times 1 \times M$. The redundant channel is then abandoned. We then concatenate $\mathbf{C}_g^n$ to form the final cost volume $\mathbf{C}_g$ in size of $H \times W \times GM$.

We input the cost volume $\mathbf{C}^1$ with feature $\mathbf{F}_{ref}^1$ to the coarse net. For the refinement net, we additionally input the up-sampled depth map result from last pyramid level. The sub-networks have the same number of layers. Each of them is encoder-decoder based network, and skip connections are applied to the layers with same resolution in the encoder and the decoder, formed the network as a U-Net. In the encoder part, the feature map is processed with convolutional layers and the resolution reduced every two layers. For decoder, the feature map is up-sampled with deconvolutional layer. Each deconvolutional layer is followed by a convolutional layer. As a result, each sub-network has 14 convolutional/deconvolutional layers. The sub-network at higher level of pyramid, e.g. the coarse net at the top level, has input features and cost volume with smaller resolution. To ensure the capability of the coarse net to capture the global depth, we employ the sub-network with more channel numbers. Since the resolution is small at top level of pyramid, large number of feature channels will not reduce the efficiency. With the resolution of feature map enlarged at lower level of pyramid, we reduce the channel numbers of the refinement network to guarantee the computation speed. Based on the previous depth result from higher level, the refinement net only needs to capture the detailed local information, thus the reduction in feature channels will not lead to loss of accuracy. As shown in Fig. 5, the depth map is estimated from coarse to fine.

C. Tracking procedure

The tracking process is based on the same feature pyramid as depth estimation network. As proposed in BA-Net [18], we further process these features with additional convolutional layers. Specifically, the feature $\mathbf{F}^n$ is added with up-sampled feature of $\mathbf{F}^{n-1}$. Then the feature map is convolved

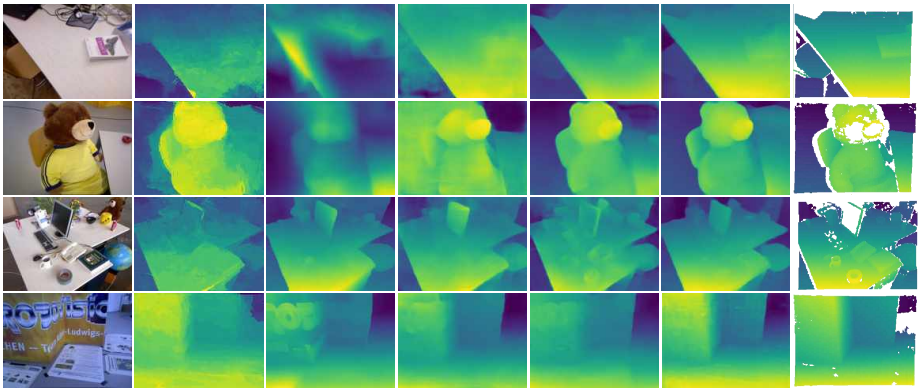

Fig. 7. Visualization results of multi-view depth estimation result from DeepMVS, DeepTAM, MVDepthNet, DENAO and ours.

TABLE I
COMPARISON RESULTS OF DEPTH ESTIMATION ON TUM DATASET.

Method	Error			Accuracy		
	L1-rel	L1-inv	sc-inv	δ_1	δ_2	δ_3
DeepMVS	0.17	0.09	0.198	0.795	0.908	0.97
DeepTAM	0.115	0.072	0.15	0.866	0.953	0.981
MVDepthNet	0.101	0.066	0.137	0.874	0.951	0.981
DENAO	0.093	0.063	**0.121**	0.891	0.959	0.979
Ours	**0.089**	**0.055**	0.126	**0.904**	**0.968**	**0.988**

with another convolutional layer, resulting in the feature $\overline{\mathbf{F}}^n (n = 1, 2, 3)$ for camera tracking.

To determine the camera pose between the current frame $\mathbf{I}_i^c$ and its reference keyframe $\mathbf{I}_j^k$, we first initialize the current camera pose $\mathbf{T}_i$ with the previous frame $\mathbf{T}_{i-1}$, then we optimize the camera pose iteratively with bundle adjustment. To this end, we transform the camera pose $\mathbf{T}_i$ into Lie algebra format as ξ_i. We re-project the feature $\overline{\mathbf{F}}_i^n$ to keyframe $\overline{\mathbf{F}}_j^n$ as $\overline{\mathbf{F}}_i^{n'}$ with (4). We sample 4096 points at $[u, v, c]^\intercal$ in high gradient region and calculate its photometric error $e_{u,v,c}^n(\xi))$ between $\overline{\mathbf{F}}_j^n$ and $\overline{\mathbf{F}}_i^{n'}$. Assuming the size of feature map $\overline{\mathbf{F}}_j^n$ is $H \times W \times GM$, the overall feature-metric error can be expressed as,

$$\mathbf{E}(\xi) = \{e_{u,v,c}^n(\xi) | u \in [0, H), w \in [0, W), c \in [0, GM)\} \tag{7}$$

Following Levenberg-Marquardt (LM) algorithm, we solve for an optimal update pose $\Delta\xi$ with:

$$\Delta\xi = (\mathbf{J}(\xi)^\intercal \mathbf{J}(\xi) + \lambda \mathbf{D}(\xi))^{-1} \mathbf{J}(\xi)^\intercal \mathbf{E}(\xi) \tag{8}$$

Where ξ is a parameterization of the lie algebra of $\mathbb{SE}(3)$, $\mathbf{J}(\xi)$ is the Jacobian matrix, $\mathbf{J}(\xi)^\intercal \mathbf{J}(\xi)$ is the Hessian matrix, $\mathbf{D}(\xi)$ is the diagonal matrix of Hessian matrix and λ is the damping factor. Then the camera pose is then updated iteratively with:

$$\xi^* = \Delta\xi \circ \xi \tag{9}$$

and $\circ$ denotes parameter update. We implement these matrix computations in deep learning framework and form up a differentiable BA module. At each pyramid level, we iteratively update the camera pose then input the result to the next level for further refinement (See Fig. 6).

D. Implement details

Our network was implemented with Tensorflow, trained and evaluated on a single Nvidia TITAN Xp GPU with 12GB of VRAM. We used the ScanNet [33] dataset to train our network. We collect image pairs and image sequences for depth network training. The image pairs are chosen from the dataset if the overlapping ratio between projected image and reference image is greater than 65%. For image sequences, we choose five frames at intervals of 2 frames. With these strategies, we collected 309k image pairs and 227k image sequences. During training, we online augment the training reference images and the corresponding neighbor images by randomly changing their brightness, saturation, hue, and randomly flipping image pairs vertically or horizontally as well as their corresponding ground truth. Accordingly, the relative camera pose is adjusted according to the flip operation.

We use the Adam optimizer during the entire training procedure. We first train the depth network and feature pyramid with image pairs for 2200k iterations, then input the image sequence to train the network for 100k iterations. To train tracking net feature, we first fix the feature of feature pyramid and train the tracking net for 100k iterations. Finally, the tracking and mapping networks are jointly trained for 10k iterations. The over all objective function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{depth} + \mathcal{L}_{pose} \tag{10}$$

and

$$\mathcal{L}_{depth} = \sum_{n=1}^{4} |\mathbf{D}^n - \mathbf{D}^{n*}| \tag{11}$$

$$\mathcal{L}_{pose} = \sum_{n=1}^{3} (|\mathbf{t}^n - \mathbf{t}^{n*}| + |\mathbf{r}^n - \mathbf{r}^{n*}|) \tag{12}$$

where $\mathbf{D}^n$ and $\mathbf{D}^{n*}$ are the predicted depth map and ground truth depth map at n^{th} pyramid level. $\mathbf{t}^n$ and $\mathbf{r}^n$ are camera translation and rotation expressed in Euler angle format, $\mathbf{t}^{n*}$ and $\mathbf{r}^{n*}$ are the corresponding ground truth.

TABLE II

COMPARISON RESULTS OF CAMERA TRAJECTORY ON TUM DATASET.

Sequence	RPE-RMSE[m/s]						ATE-RMSE[m]					
	init w/ gt			init w/o gt			init w/ gt			init w/o gt		
	LSD-SLAM	DeepTAM	Ours	LSD-SLAM	DeepTAM	Ours	LSD-SLAM	DeepTAM	Ours	LSD-SLAM	DeepTAM	Ours
fr1/desk	0.960	0.121	**0.075**	0.237	0.120	**0.073**	2.124	0.336	**0.230**	0.571	0.337	**0.186**
fr1/xyz	**0.024**	0.033	0.088	0.049	**0.032**	0.064	**0.015**	0.051	0.093	**0.031**	0.048	0.090
fr1/360	crash	0.127	**0.106**	crash	0.114	**0.093**	crash	0.223	**0.128**	crash	0.205	**0.089**
fr1/desk2	19.238	0.201	**0.163**	0.442	0.217	**0.143**	23.613	0.488	**0.399**	0.761	0.584	**0.294**
fr1/floor	0.286	0.282	**0.217**	0.286	0.311	**0.102**	0.764	0.629	**0.611**	0.790	0.689	**0.244**
fr1/plant	0.212	0.281	**0.098**	**0.069**	0.297	0.112	0.351	0.776	**0.210**	**0.096**	0.667	0.304
fr1/room	0.727	0.150	**0.123**	0.335	0.157	**0.116**	1.002	0.636	**0.298**	0.639	0.637	**0.285**
fr1/rpy	0.110	**0.039**	0.069	0.063	**0.040**	0.043	**0.060**	0.065	0.090	**0.053**	0.078	0.056
fr1/teddy	0.240	0.184	**0.110**	0.303	0.173	**0.132**	0.670	0.444	**0.310**	0.774	0.376	**0.312**
average	2.725	0.158	**0.117**	0.223	0.162	**0.098**	3.575	0.405	**0.263**	0.464	0.402	**0.207**

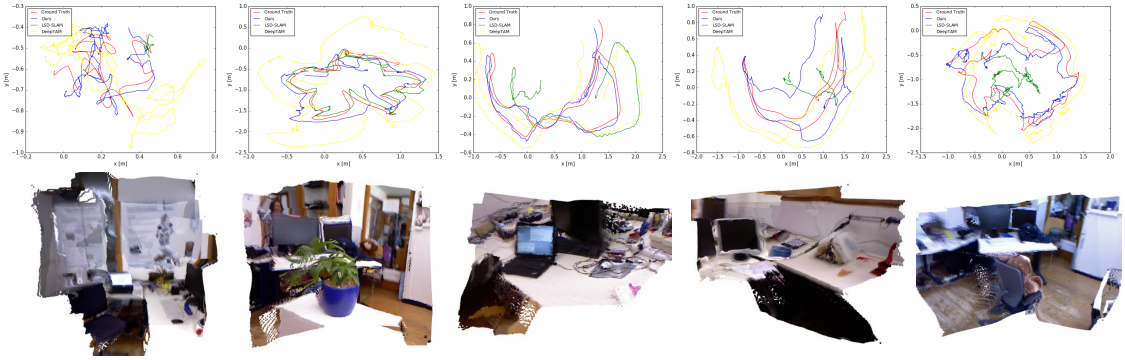

Fig. 8. The visualization results of camera trajectory and reconstructed depth structure on fr1/360, fr1/plant, fr1/desk, fr1/desk2 and fr1/teddy.

During network training, we warp features and build cost volume with ground truth camera pose. The inverse depth range of coarse net d_{min}, d_{max} are set to 0.01 and 2.5 respectively. The pose is updated for 3 times at each pyramid level during training. For deployment, we update camera pose for 10 times at each pyramid level in BA module.

IV. EXPERIMENT

We use 9 sequences in TUM dataset [34] to evaluate the performance of our D^{2}VO. We first evaluate the depth estimation accuracy of our mapping network. To compare with other methods, we construct our test set as follows. In each image sequence, we uniformly sampled five images with a stride of 10 images in a non-overlapping sliding window manner. We treat the 3^{rd} image of every five sampled images as the reference image, and the other four images as the neighbor images. As a result,we obtain a total of 337 sets of test image sets. We use four metrics to measure the error and accuracy of the depth result, i.e. relative error (L1-rel), inverse depth error (L1-inv), scale invariant error (sc-inv) and percentage of predicted pixels where the L1-rel is within a threshold δ. We compare our method with four state-of-the-art multi-view depth estimation methods, i.e. DeepMVS [28], MVDepthNet [10], DENAO [30], DeepTAM [12]. The result of those methods are obtained with their source codes and pre-trained models.

As shown in Table. I and Fig. 7, we outperforms the state-of-the-art methods on both quantity and quality for all the evaluation metrics. It is worth mentioning that the training dataset and testing dataset of our network are entirely different, while DeepMVS, MVDepthNet and DENAO used sampled image pairs from TUM dataset for training. Some scenarios in their training data and test data may be quite similar. Compare with one-stage depth prediction network MVDepthNet and DeepMVS, our depth network refines the depth from coarse to fine with muliple sub-networks. The mapping networks of both D^{2}VO and DeepTAM are built based on this coarse-to-fine and iterative optimization strategy. However, D^{2}VO predicts depth with hierarchical network and estimates depth from low resolution. DeepTAM inputs the depth result iteratively into the same network with fixed resolution.

To evaluate the tracking accuracy of D^{2}VO, we compare our system with traditional direct method based SLAM, i.e. LSD-SLAM [2], and learning based tracking and mapping method, i.e. DeepTAM [12], on 9 sequences from TUM dataset. The trajectory results are evaluated with RPE and ATE error. To remove the affect of initialization method, we first initialize all the systems with ground truth depth map (i.e. init w/ gt) and compare the tracking result. Then we utilize the separate initialization method of each system (i.e. init w/o gt) and compare the result of the full system. The comparison results are shown in Table. II. We outperform the other methods on 6 sequences in 9 sequences of TUM dataset. The comparison results between different initialization methods demonstrate the effectiveness of our

initialization strategy. In Fig. 8, we display the trajectory and reconstruction results of five example image sequences.

V. CONCLUSIONS

Geometric information is essential for visual odometry. To integrate geometric cues into both tracking and mapping of learning based VO system, we propose D^{2}VO, which estimate depth of keyframes with multi-view depth estimation network and track camera poses with direct method. The light-weighted depth net is designed with a hierarchical coarse-to-fine strategy, guaranteeing the efficiency and accuracy of our mapping process. The system initialization is based on the same depth network, which saves the memory space of the program. The camera poses are calculated by minimizing the photometric error between feature maps of the frames. The tracking and mapping are tightly coupled with keyframes as traditional VO/SLAM systems. Benefiting from both traditional geometric calculation and deep CNN, our D^{2}VO yields state-of-the-art results on both tracking and mapping. In the future, we will attempt to extend our system to joint optimize the depth and pose with CNN.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[3] 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*, 2014, pp. 2366–2374.

[4] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[5] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.

[6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011.

[7] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756.

[8] H. Luo, Y. Gao, Y. Wu, C. Liao, X. Yang, and K.-T. Cheng, "Real-time dense monocular slam with online adapted depth prediction network," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 470–483, 2018.

[9] X. Yang, Y. Gao, H. Luo, C. Liao, and K.-T. Cheng, "Bayesian denet: Monocular depth prediction and frame-wise fusion with synchronized uncertainty," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2701–2713, 2019.

[10] K. Wang and S. Shen, "Mvdepthnet: real-time multiview depth estimation neural network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 248–257.

[11] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 767–783.

[12] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 822–838.

[13] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[14] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[15] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[16] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

[17] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2022–2030.

[18] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint arXiv:1806.04807*, 2018.

[19] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[20] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[21] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[22] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[23] Z. Yuan, D. Zhu, C. Chi, J. Tang, C. Liao, and X. Yang, "Visual-inertial state estimation with pre-integration correction for robust mobile augmented reality," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, p. 1410–1418.

[24] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6243–6252.

[25] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, "Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5218–5223.

[26] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from rgb and sparse sensing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 167–182.

[27] X. Yang, J. Chen, Z. Wang, Q. Zhang, W. Liu, C. Liao, and K.-T. Cheng, "Monocular camera based real-time dense mapping using generative adversarial network," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 896–904.

[28] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2821–2830.

[29] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[30] J. Chen, Q. Jia, and C. Liao, "Denao: Monocular depth estimation network with auxiliary optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 02 2020.

[31] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[32] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3273–3282.

[33] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.