Deep Unsupervised Visual Odometry Via Bundle Adjusted Pose Graph Optimization

Guoyu Lu

Abstract— Unsupervised visual odometry as an active topic has attracted extensive attention, benefiting from its label-free practical value and robustness in real-world scenarios. However, the performance of camera pose estimation and tracking through deep neural network is still not as ideal as most other tasks, such as detection, segmentation and depth estimation, due to the lack of drift correction in the estimated trajectory and map optimization in the recovered 3D scenes. In this work, we introduce pose graph and bundle adjustment optimization to our network training process, which iteratively updates both the motion and depth estimations from the deep learning network, and enforces the refined outputs to further meet the unsupervised photometric and geometric constraints. The integration of pose graph and bundle adjustment is easy to implement and significantly enhances the training effectiveness. Experiments on KITTI dataset demonstrate that the introduced method achieves a significant improvement in motion estimation compared with other recent unsupervised monocular visual odometry algorithms.

I. Introduction

Visual odometry is an essential task in robotics and computer vision to simultaneously determine the camera pose and recover 3D structures from sequential images. It also enables a wide range of applications on augmented reality [25] [28], unmanned aerial vehicle (UAV) [3] [30], and self-driving cars [38] [5].

In the last decade, visual odometry and SLAM systems have been developed both from front-end to track camera motion in real-time and from back-end to locally and globally optimize the 3D structures and camera motion, which have achieved promising and robust performance. However, the conventional systems still rely on traditional image features for detection and matching, which frequently fail in challenging environments, such as diverse lighting and exposure conditions, no or repeated textures, and large portion of moving foreground. Moreover, direct methods for dense reconstruction [37] [32] cannot efficiently optimize the entire depth image in real-world applications due to the extremely large size of parameters and variables for updating.

Recent deep neural network based camera pose estimation and 3D scene recovering algorithms [34] [16] [35] [1] [4] [39] are able to achieve better performance than the conventional pipelines in scene depth estimation, in terms of both point cloud density and accuracy. However, unsupervised learning based network still cannot achieve comparable results with conventional geometric based visual odometry pipelines in camera motion estimation, due to the lack of efficient and effective pose drift and map optimization, frequently leading to large gaps in loop closures. So, it is necessary to incorporate online optimization of pose graph and bundle adjustment into an unsupervised deep neural network to explore the benefits from both conventional VO/SLAM algorithms and deep neural network-based methods.

Guoyu Lu is with the Intelligent Vision and Sensing Lab at the University of Georgia, USA guoyu.lu@uga.edu

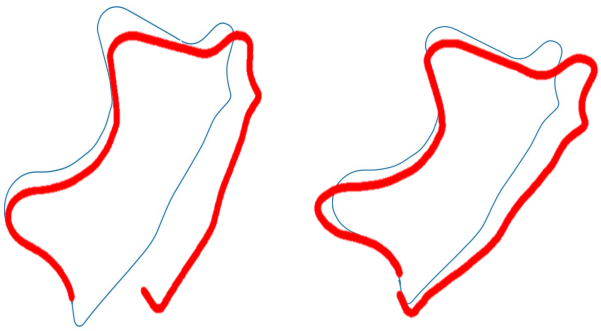

Fig. 1: Trajectory comparison without (left) and with (right) the proposed bundle adjusted pose graph. Blue: ground truth plotting. Red: camera trajectory estimated from neural networks.

In this work, we propose a pose graph and bundle adjustment optimization embedded deep visual odometry network, namely PBO-VONet, to learn the optimized camera pose and dense depth at the same time, realizing bundle adjustment on both dense depth and camera motion estimation during training. With the bundle adjustment and pose graph optimization module, we are able to prevent pose drift effectively and improve the performance of unsupervised deep learning based visual odometry by a large margin. The proposed framework bridges the gap between the conventional algorithms and deep learning based networks by leveraging both benefits of them via local and global optimization, photometric consistency, and geometric-consistency. The sample outputs of our designated framework are depicted in Fig. 1.

To summarize, the main contributions of this work are: i) We introduce the pose graph and bundle adjustment as an online optimization into the deep visual odometry network, enabling continuously updating dense depth and camera motion to be practicable. To the best of our knowledge, this is one of the first attempts to embed both pose graph optimization and bundle adjustment as online optimization methods to the unsupervised deep learning network. ii) Both trajectory estimation and 3D mapping from our introduced algorithm outperform existing approaches largely. iii) The bundle adjustment process is seamlessly integrated into the neural network in real applications, bringing effective enhancement for camera pose estimation that is deficient in most deep neural network based VO methods.

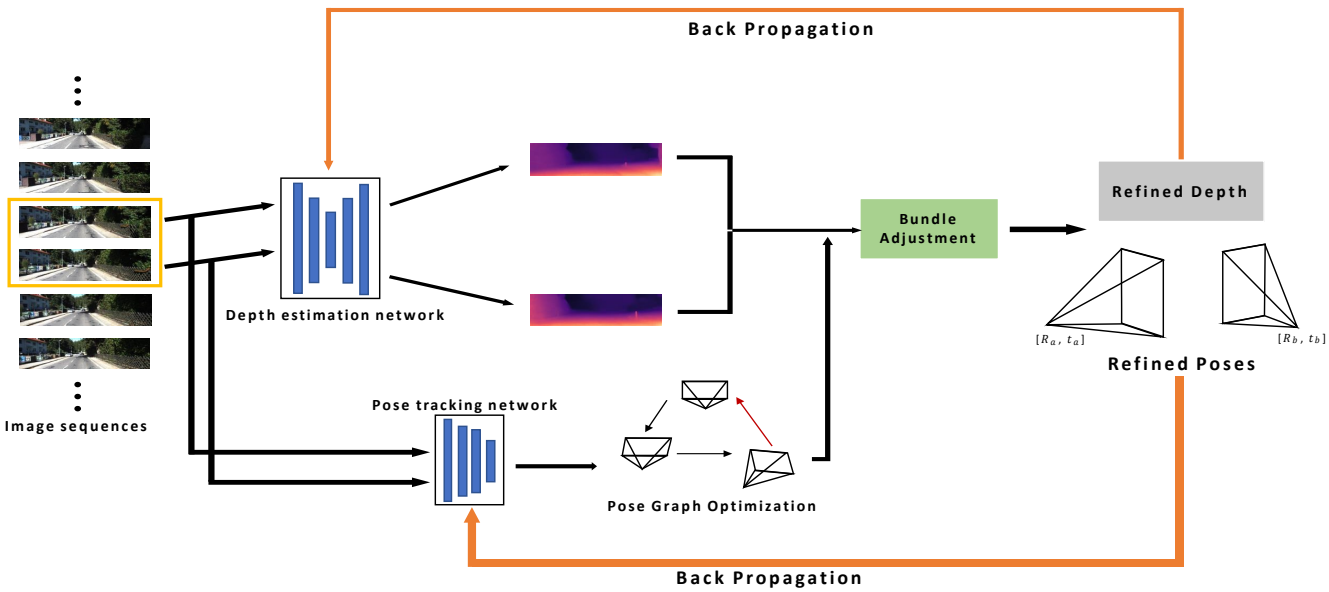

Fig. 2: An architecture overview of the proposed pose graph and bundle adjustment optimized deep visual odometry framework. The framework will output both depth maps and camera poses which compose a pose graph for further optimization in bundle adjustment with depth values of salient pixels.

II. Related Work

There is a large body of work exploring the idea of estimating both dense depth and camera motion from monocular video inputs. We will discuss the conventional geometric visual odometry methods, deep learning based visual odometry algorithms, and a combination of them as below.

Conventional VO approaches. In order to estimate motion, early conventional feature-based VO approaches rely on the extracted geometric constraints from image [26]. More specifically, these approaches can be concluded into two branches: featured based approaches and direct approaches [10] [36]. Feature based approaches require feature tracking [19], which usually fall into drifting when considering the accumulation of time and only some of the features in the image are extracted during the calculation, leaving a large amount of information in the original image unused [31]. Direct approaches [9] then put forward to solve the problem by going through all the pixels in the image sequential. These approaches take the advantage of whole-image information which contributes to higher accuracy, but suffer from large frame-to-frame motion. Another shortcoming is that these approaches occupy too much computation due to the heavy optimization burden for dense tracking. With maps available, image sequences are also used for estimating the camera poses [22] [20] [21] [18].

Deep learning based networks for VO. Deep neural networks have been applied in visual odometry to recover both trajectory and scene depth simultaneously [38] [34] [31] [24] [40] [37] [16] [29] [2] [33]. Among them, Wang et al. [31] trained a Recurrent Neural Network (RNN) for estimating the camera poses and scene depth. Zhou et al. [37] designed a coarse-to-fine strategy to track camera pose from keyframes. However, the aforementioned methods all require a large amount of ground truth poses for training. To militate the need of costly labels, unsupervised methods have attracted much attention recently. Li et al. [16] extended [31] to design an unsupervised learning framework to use spatial and temporal information extracted from stereo sequences as constraints. Yin et al. [34] simultaneously learned depth, camera pose estimation, and optical flow via a CNN network. Ranjan et al. [29] further integrated depth estimation, camera motion, optical flow, and segmentation in a unit framework during the training. Though these systems are capable of achieving a good estimation of the scene depth, camera pose estimation is still not as accurate as conventional pipelines due to the lack of back-end optimization modules.

Combination of conventional VO and deep learning based VO. Considering the superior performance of recent deep learning based networks in feature detection and representation, a few works [17] [12] explored integrating image features learned from deep learning networks into the conventional visual odometry pipeline. Li et al. [17] introduced a hybrid VO system to combine deep learning based monocular VO algorithms with a windowed pose graph as an additional back-end. A stereo VO approach SuperPointVO [12] was proposed to replace traditional feature detection with a CNN-based feature extraction network SuperPoint [7], and integrate it into a standard stereo visual odometry system. Although these methods can leverage the benefits from both conventional VO pipelines and deep learning based networks, they still lack strict geometric optimization, e.g., bundle adjustment. Also, the pre-trained feature extraction network might not perform well in different and unseen scenarios.

III. Bundle Adjusted VO Framework

The proposed PBO-VONet is comprised of a set of key components: an unsupervised monocular VO pipeline based on both geometric and photo-metric consistencies across

local neighboring frames, a graph-based pose optimization module and a pose-depth bundle adjusted optimization. To enable an efficient and practicable optimization, we propose to update only selected keypoints in the depth map in the optimization process, while inferring the entire dense depth. The use of all image pixels for optimization would result in the difficulty of convergence of model training due to the significant parameters to optimize (e.g., optimize hundreds of thousands images with over one hundred thousand pixels for each image). To the best of our knowledge, our proposed network is one of the first approaches to enable online optimization in the unsupervised deep VO structure. An overview of our training pipeline is depicted in Fig. 2.

A. Unsupervised Monocular VO Pipeline

Given monocular video sequences, we are able to use geometric and photometric consistencies between the target frame to reference views to train depth estimation and motion estimation. As illustrated in Fig. 2, the self-supervision simultaneously constrains the depth inference network and pose estimation network. Pose estimation network is trained by multiple adjacent local frames composed of a target frame I_t and the referenced neighboring frames I_{t+1}. A group of relative poses are able to be inferred. Simultaneously, corresponding depth map for each input frame is generated by the depth estimation network. The initial estimated depth maps and pose vectors will then be optimized by the pose graph and bundle adjustment, which will be detailed in Sec. III-B and Sec. III-C.

1) Multi-view Re-projection Loss: Given each pair of two images I_t and I_{t+1}, the estimated depth map D_t, and the estimated camera motion $T_{t->t+1}$, we are able to compute the per-pixel correspondence by projecting the pixel of the target image to the reference images. Supposing a known camera intrinsic K, the correspondence of the pixel p_t in I_{t+1} can be represented by the following equation:

$$p_{t+1} = K\tilde{T}_{t->t+1}\tilde{D}(p_t)K^{-1}p_t \tag{1}$$

To warp the target frame I_t to reference frame I_{t+1} and constrain a smooth reconstruction $\tilde{I}_{t+1}$, we compute the per-pixel minimum photometric loss across multiple reference frames rather than the averaging photometric error [11] as:

$$L = \sum_{i=1}^{N} \min_{t' \in \{t-i,t+i\}} \rho(I_t, I_{t'->t}) \tag{2}$$

where N is the number of frames. ρ is a weighted combination of L1 loss term and the structural similarity index measure (SSIM) loss [14] to achieve a robust image reconstruction performance, denoted as:

$$\rho(I_1, I_2) = \frac{\alpha}{2}(1 - SSIM(I_1, I_2)) + (1-\alpha)||I_1 - I_2||_1 \tag{3}$$

2) Moving Object Masking: As the loss constraint Eq. 2 should meet the assumption of static scenes and moving cameras, objects with large motion and occlusions will create non-rigid transformation which will degrade the learning effect of camera pose and depth estimation. In this case, we propose to incorporate the depth inconsistency mask [4] to exclude the moving objects and regions. The depth inconsistency map for each pixel value p is computed as:

$$D_{diff}(p) = \frac{|D^t_{t+1}(p) - D'_{t+1}(p)|}{D^t_{t+1}(p) + D'_{t+1}(p)} \tag{4}$$

where D^t_{t+1} is the synthesized depth at $t+1$ frame generated from I_t based on the estimated camera motion $T_{t->t+1}$, and D'_{t+1} is the bilinear interpolation of the estimated depth at $t+1$ frame. So, the moving mask can be computed based on the depth inconsistency map D_{diff} as:

$$M_{moving} = 1 - D_{diff}(p) \tag{5}$$

where M_{moving} ranges from 0 to 1, which intends to give small weights to the regions containing moving and occluded objects. Considering that there could exist non-moving frames in specific scenes (stopping), which may affect the training of camera motion estimation, we apply auto-masking to compute the photometric loss between the neighboring moving frames only, filtering out those points whose relative motion is the same:

$$M_{auto} = \begin{cases} 1, & \text{if } ||I_t - I'_t||_1 < ||I_t - I_{t+1}||_1 \\ 0, & \text{else} \end{cases} \tag{6}$$

where M_{auto} is a binary mask. I'_t is a warped frame from I_{t+1} based on the estimated depth map $\tilde{D}$ and relative camera motion $\tilde{T}$.

B. Pose Graph Optimization

Normally, pose estimation from the deep neural network suffers from a relatively large drift. We propose to incorporate pose graph to optimize each camera pose node $c = [\tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_n]$ computed from the estimated relative rigid camera transformation $T = [\tilde{T}_1, \tilde{T}_2, ..., \tilde{T}_n]$. Let $z_{ij} = \gamma(\tilde{c}_i, \tilde{c}_j) + n_{ij}$ to be the edge of each camera pose vertex pair $\tilde{c}_i$ and $\tilde{c}_j$, where the noise is formulated as a zero-mean white Gaussian as $n_{ij} \sim N(0, W_{ij})$. The graph optimization is then described as a problem of maximizing the posterior probability of all points on the camera's trajectory, given the estimated camera pose c and the observed edge constraints γ between the pose nodes:

$$Prob(Z|c) = \prod_{ij} prob(z_{ij}|(\tilde{c}_i, \tilde{c}_j)) \tag{7}$$

By following the Gaussian distribution assumption and taking the natural logarithm on both sides of Eq. 7, the maximum likelihood estimation can be easily converted to the minimization problem by the following least-square function:

$$\begin{aligned} \tilde{x} &= \operatorname{argmin}_x \sum_{ij} e_{ij}^T \Sigma_{ij} e_{ij} = \\ &\operatorname{argmin}_x \sum_{ij} ||z_{ij} - \gamma(\tilde{c}_i, \tilde{c}_j)||^T \Sigma_{ij}^{-1} ||z_{ij} - \gamma(\tilde{c}_i, \tilde{c}_j)|| = \\ &\operatorname{argmin}_x \sum_{ij} ||z_{ij} - \gamma(\tilde{c}_i, \tilde{c}_j)||^T W^{-1} ||z_{ij} - \gamma(\tilde{c}_i, \tilde{c}_j)|| \end{aligned} \tag{8}$$

where Eq. 8 is a non-linear least-square optimization and e_{ij} is the error between z_{ij} and the estimated value $\gamma(\tilde{c}_i, \tilde{c}_j)$.

To solve the optimization equation, iterative Gauss-Newton is used for solving Eq. 8. Specially, an optimization for the estimated camera pose $\tilde{c}^{(n)}$ at the current time n is calculated by the approximation of the second-order Taylor-series as:

$$\tilde{c} = \sum_{ij} ||z_{ij} - \gamma(\tilde{c}_i^{(n)}, \tilde{c}_j^{(n)}) - \Gamma_{ij}^n \delta c||^T W^{-1} \\ ||z_{ij} - \gamma(\tilde{c}_i^{(n)}, \tilde{c}_j^{(n)}) - \Gamma_{ij}^n \delta c|| = ||J^{(n)}\delta c - k^{(n)}||^2 \tag{9}$$

where k is the corresponding residual vector as equation below, and Γ_{ij}^n is the partial derivative of the edge constraint γ to the estimated camera pose $\tilde{c}$, and J is the Jacobian matrix which is composed of all the computed Jacobians Γ as:

$$J = \begin{bmatrix} W_{12}^{-1/2}\Gamma_{12} \\ \cdots \\ W_{i-1,j-1}^{-1/2}\Gamma_{i-1,j-1} \\ W_{i,j}^{-1/2}\Gamma_{i,j} \end{bmatrix}$$

$$k = \begin{bmatrix} W_{12}^{-1/2}(z_{12} - \gamma(\tilde{c}_1, \tilde{c}_2)) \\ \cdots \\ W_{i-1,j-1}^{-1/2}(z_{i-1,j-1} - \gamma(c_{i-1}, c_{j-1})) \\ W_{i,j}^{-1/2}(z_{i,j} - \gamma(\tilde{c}_i, \tilde{c}_j)) \end{bmatrix} \tag{10}$$

Eq. 9 can be further simplified by applying QR factorization on J. Hence, Eq. 9 can be rewritten as:

$$\min||J\delta c - k||^2 = ||[Q_1 \;\; Q_2] \begin{matrix} R \\ 0 \end{matrix} \delta c - k||^2 = \\ ||Q_1 \begin{matrix} R \\ 0 \end{matrix} \delta c - k||^2 = || \begin{matrix} R \\ 0 \end{matrix} \delta c - Q_1^T k||^2 = \\ || \begin{matrix} R \\ 0 \end{matrix} \delta c - Q_1^T k||^2 = \min||R\delta c - d||^2 \tag{11}$$

Hence, δc can be computed as:

$$\delta c = (R^T R)^{-1} R^T d \tag{12}$$

Based on the pose graph optimization, the optimized camera pose $\tilde{c}_{update}$ can be corrected from the initial estimation from the pose estimation network $\tilde{c}$ and a small correction δc as: $\tilde{c}_{update} = \tilde{c} + \delta c$. Hence, the relative pose estimation can be correspondingly refined to $\tilde{T}_{update} = \tilde{T} + \delta T$.

C. Bundle Adjustment Integration

Considering that the pose graph optimization ignores the 3D point information and the self-supervision from the unsupervised VO network is able to constrain both initial scene depth $\tilde{D}$ and the refined relative camera pose $\tilde{T}_{update}$, we propose to further refine them for more precise poses and depths by solving them in geometric bundle adjustment (BA) optimization. This process is formulated as minimizing the total energy E of the re-projection errors e on the image pixel p across all the frames as:

$$E = \operatorname{argmin} \sum_{i=1} \sum_{j=1} ||e_{ij}(p, \tilde{T}_{update}, \tilde{D})|| = \\ \operatorname{argmin} \sum_{i=1} \sum_{j=1} ||I(p_{i,j}) - I_i(\pi(\tilde{T}_{update,i}, M(\tilde{D}_j)))|| \tag{13}$$

where the global energy E that needs to be minimized is composed by a series of errors between the pixel intensity of the projected 3D points and the corresponding image pixel. Considering that it is not practicable to optimize the entire depth estimated from the depth estimation network, we only selected 2000 keypoints (ORB feature is used in our setting) from the input image.

To minimize the global energy E over all depths at the selected keypoints and the corresponding camera motion, we define the parameter vector P and the measurement vector X as:

$$P = (\tilde{T}_{update,1}^T, \ldots, \tilde{T}_{update,i}^T \mid \\ M(\tilde{D}_1), M(\tilde{D}_2), \ldots, M(\tilde{D}_j))^T, \\ X = (p_{11}^T, p_{12}^T, \ldots, p_{21}^T, \ldots p_{ij}^T)^T \tag{14}$$

The estimated measurement vector $\hat{X}$ can be expressed as:

$$\hat{X} = (\hat{p}_{11}^T, \hat{p}_{12}^T, \ldots, \hat{p}_{1m}^T, \hat{p}_{21}^T, \ldots, \hat{p}_{nm}^T) = v(P + \Delta) = \\ v(\tilde{T}_{update} + \delta\tilde{T}_{update}, M(\tilde{D}) + \delta M(\tilde{D})) \approx \\ v(P) + A\delta\tilde{T}_{update} + B\delta M(\tilde{D}) \tag{15}$$

Therefore, the bundle adjustment optimization is equal to minimize the squared Σ_X^{-1} norm as:

$$\epsilon^T\epsilon = \sum_i \sum_j ||\epsilon_{ij}||^2 = ||X - \hat{X}||^2 \rightarrow \epsilon^T \Sigma_X^{-1} \epsilon \\ = ||X - \hat{X}||^2 \Sigma_X \tag{16}$$

Σ represents convariance matrix. The above normal equation can be solved with Levenberg–Marquardt (LM) non-linear least-square algorithm:

$$(J^T \Sigma_X^{-1} J + \mu I)\delta = J^T \Sigma_X^{-1} \epsilon \tag{17}$$

The updating vector for LM algorithm becomes:

$$\delta = (\delta_{\tilde{T}_{update}}, \delta_{M(\tilde{D})}^T) = (\delta_{\tilde{T}_1}^T, \delta_{\tilde{T}_2}^T, \ldots, \\ \delta_{\tilde{T}_i}^T, \delta_{M(\tilde{D}_1)}^T, \delta_{M(\tilde{D}_2)}^T, \ldots, \delta_{M(\tilde{D}_j)}^T) \tag{18}$$

And the Jacobian matrix J is:

$$\frac{\partial \hat{X}}{\partial P} = [\frac{\partial \hat{X}}{\partial \tilde{T}_{update}} | \frac{\partial \hat{X}}{\partial M(\tilde{D})}] \tag{19}$$

Therefore, the covariance matrix becomes:

$$\Sigma_X = diag(\Sigma_{X_{11}}, \Sigma_{X_{12}} \ldots \Sigma_{X_{ij}}) \tag{20}$$

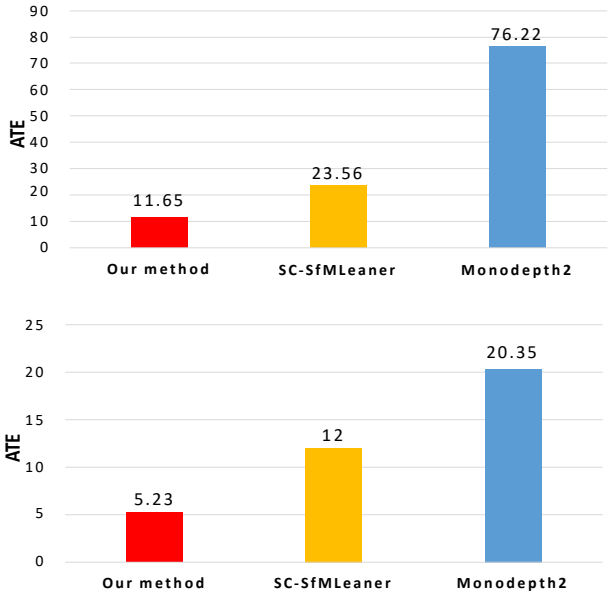

Fig. 3: Absolute Trajectory Error (ATE) results on the KITTI sequence 09 (top) and 10 (bottom) in comparison with SC-SfMLearner [4] and Monodepth 2 [11].

With that, the normal equation takes the form:

$$\begin{bmatrix} A^T A & A^T B \\ B^T A & B^T B \end{bmatrix} \begin{pmatrix} \delta \tilde{T}_{update} \\ \delta M(\tilde{D}) \end{pmatrix} = \begin{pmatrix} A^T \epsilon_{\tilde{T}_{update}} \\ B^T \epsilon_{M(\tilde{D})} \end{pmatrix} \rightarrow \begin{bmatrix} U^{\square} & W \\ W^T & V^{\square} \end{bmatrix} \begin{pmatrix} \delta \tilde{T}_{update} \\ \delta M(\tilde{D}) \end{pmatrix} = \begin{pmatrix} A^T \epsilon_{\tilde{T}_{update}} \\ B^T \epsilon_{M(\tilde{D})} \end{pmatrix} \tag{21}$$

δa can be formulated as:

$$(U^{\square} - WV^{\square -1} W^T)\delta_{\tilde{T}_{update}} = \epsilon_{\tilde{T}_{update}} - WV^{\square -1}\epsilon_{M(\tilde{D})} \rightarrow \delta_{\tilde{T}_{update}} = (U^{\square} - WV^{\square -1} W^T)^{-1}\epsilon_{\tilde{T}_{update}} - WV^{\square -1}\epsilon_{M(\tilde{D})} \tag{22}$$

Then δb can be solved to optimize the 3D points by solving:

$$V^{\square}\delta_{M(\tilde{D})} = \epsilon_{M(\tilde{D})} - W^T \delta_{\tilde{T}_{update}} \tag{23}$$

Therefore, the final refined camera pose estimation and depth map can be expressed by $\tilde{T}_{final}$ and $\tilde{D}_{final}$, respectively.

IV. Experimental Results

A. Implementation Details

Dataset: We separately evaluate the depth estimation and odometry on the Eigen split of KITTI raw dataset and KITTI odometry dataset. We first evaluate the depth estimation performance on the Eigen testing split [8], with all images resized to 832 × 256. For KITTI odometry dataset, we follow the standard setting to use sequences 00-08 for training and 09-10 for testing.

Network architecture and training: The designated framework is implemented on PyTorch [27] with a single Nvidia P6000 GPU. ResNet-18 [13] is applied as the encoder backbone to generate the depth map in the unsupervised VO network. The depth decoder contains sigmoid activation functions at the output and ELU [6] as the non-linear activation. The camera pose estimation network consists of 7 convolutions followed by a 1×1 convolution to output 6-DoF pose, with a ResNet-18 based extractor which is similar as [11]. The number of training epochs is 30 with a mini-batch size of 4 for our experiment. Adam optimizer [15] is applied with a learning rate of 0.0001. The images are first resized to 832×256 for both camera pose and depth estimation. The testing process runs on the same setup.

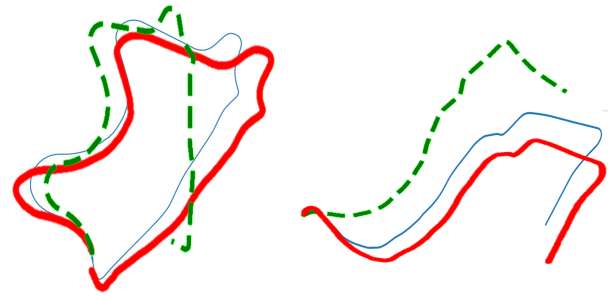

Fig. 4: Comparison of the trajectory on the KITTI odometry 09 (left) and 10 (right) splits. Blue: Ground truth; Red: Camera trajectory from our system; Green: Camera trajectory from [11].

B. Odometry Evaluation

We evaluate the proposed method on the odometry split of the KITTI dataset. 00-08 sequences are used for training and 09-10 sequences are tested, which maintains the same setting as [4] [11]. Fig. 3 depicts the odometry results on the KITTI dataset. It can be seen that our method achieves clearly improvement in the absolute trajectory error (ATE) when comparing with other methods [4] [11] under the same setup condition. Fig. 4 further shows qualitative comparison results for estimating the trajectories on the sequence 09 and 10. The plotted trajectories based on our final output relative poses are closer to the ground truth plotting than other methods, even compared to [11]. Especially, our estimated trajectory is very close to be a loop on the sequence 09, even without any loop closure detection.

C. Depth Estimation Results

We retrain our method on the raw dataset of KITTI and compare with other methods on the Eigen test set. In Fig. 5, we show some qualitative results. The proposed method is able to better preserve the object shapes and boundaries and prevent some mis-predictions in trees and the sky.

Table I makes an comparison with our model and other depth estimation approaches. Quantitative results on depth estimation are provided in Table I. Absolute relative difference (Abs Rel), squared relative difference (Sq Rel), Root Mean Square Error (RMSE), RMSE log, and the accuracy under thresholds ($\delta_i < 1.25^i$, i = 1, 2, 3) are reported. The results shown in Table I clearly shows that our depth estimation approach outperforms other approaches, achieving the new state-of-the-art performance in most metrics, especially the Abs Rel error and the first accuracy.

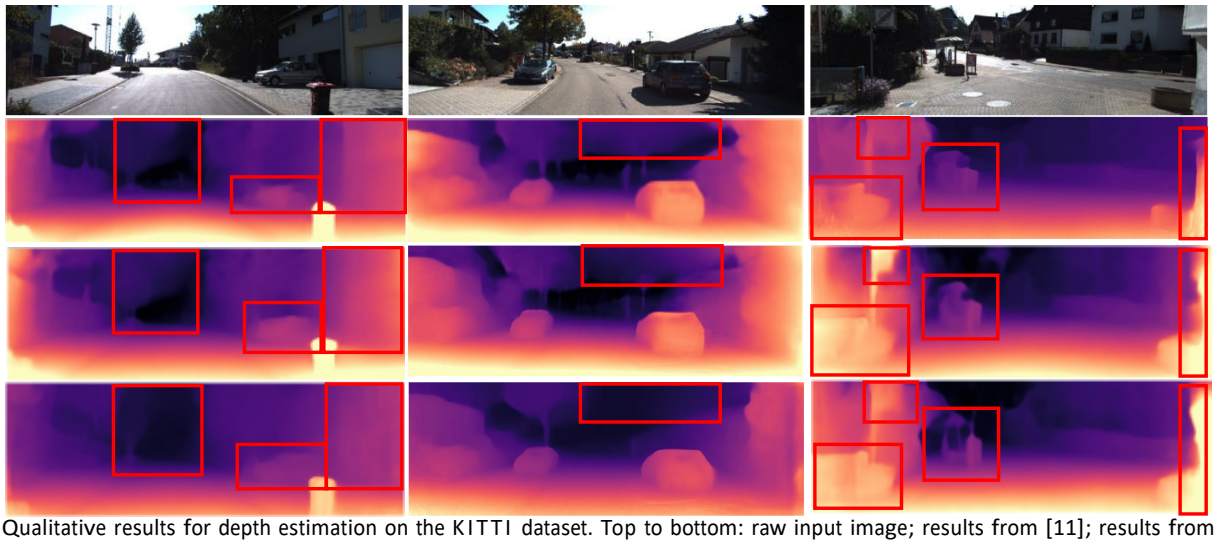

Fig. 5: Qualitative results for depth estimation on the KITTI dataset. Top to bottom: raw input image; results from [11]; results from [23]; our results. Red boxes mark the major differences.

Methods	Training type	Error				Accuracy		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
SC-SfMLearner [4]	Unsupervised	0.141	1.224	5.548	0.218	0.811	0.934	0.972
Monodepth2 [11]	Unsupervised	0.130	1.144	5.485	0.232	0.831	0.932	0.968
HR-Depth [23]	Unsupervised	0.133	1.062	5.381	0.216	0.826	0.936	0.973
Ours with bundle adjusted pose graph	Unsupervised	0.118	1.007	5.099	0.196	0.852	0.946	0.979

TABLE I: Quantitative comparison with other recent methods. All methods are trained based on KITTI Eigen training split for a fair comparison. We compare with unsupervised methods taking both a single image [11] and monocular video [4] [23] as input for testing.

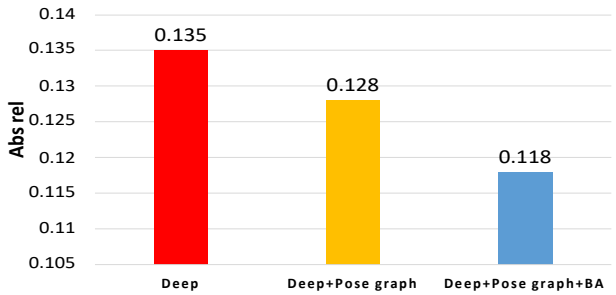

Fig. 6: Ablation study for depth estimation based on basic deep VO, deep VO+Pose graph, and deep VO+Pose graph+BA in Abs rel.

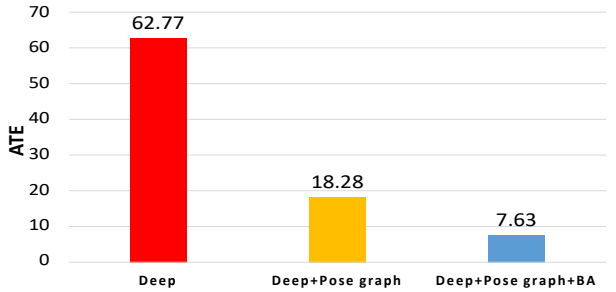

Fig. 7: Ablation study for camera pose motion estimation based on basic deep VO, deep VO+Pose graph, and deep VO+Pose graph+BA in ATE.

D. Ablation Analysis

To validate the core components introduced in our proposed network, we perform an ablation study on the KITTI Odometry dataset. Three different settings are compared as below: 1. Results directly from the deep unsupervised network (”Deep”) 2. Results from the deep unsupervised network with pose graph optimization (”Deep+Pose graph”) 3. Results from the deep unsupervised network with bundle adjusted pose graph optimization (”Deep+Pose graph+BA”) as our full pipeline. We study the effect of the mentioned key component for training. It can be observed in Fig. 6 and Fig. 7 that compared with the baseline result di-rectly from the network (”Deep”), ”Deep+Pose graph” and ”Deep+Pose graph+BA” both give the depth estimation and VO performance a significant boost. Moreover, ”Deep+Pose graph+BA” tends to perform better than ”Deep+Pose graph”, which can be explained by the fact that bundle adjustment simultaneously refine the depth and pose graph instead of focusing only on the poses.

V. Conclusion

In this paper, we propose an unsupervised monocular visual odometry method with integration of camera pose graph optimization and bundle adjustment during the training process. Pose graph and bundle adjustment optimization greatly contribute to our framework by updating the motion and depth and refining outputs, which mitigates the pose drift issues taking place in the trajectory estimation. By selecting keypoints to optimize together with the camera poses, the neural network can be effectively trained. Extensive qualitative and quantitative results on KITTI dataset demonstrate the significant improvement in trajectory estimation and depth estimation. The method benefits from both graph optimization in conventional approaches and feature learning capability of deep learning-based methods to correct pose drifting and depth miss-prediction significantly.

Acknowledgement

This publication is based upon work supported by NSF under Awards No. 2104032 and 2105257.

REFERENCES

[1] Y. Almalioglu, M. Saputra, P. de Gusmao, A. Markham, and N. Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In IEEE International Conference on Robotics and Automation (ICRA), 2019.

[2] L. Andraghetti et al. Enhancing self-supervised monocular depth estimation with traditional visual odometry. In International Conference on 3D Vision (3DV), 2019.

[3] E. Bedell, M. Leslie, K. Fankhauser, J. Burnett, M. G. Wing, and E. A. Thomas. Unmanned aerial vehicle-based structure from motion biomass inventory estimates. Journal of Applied Remote Sensing, 11(2):026026, 2017.

[4] J.-W. Bian, H. Zhan, N. Wang, Z. Li, L. Zhang, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth learning from video. International Journal of Computer Vision, 129(9):2548–2564, 2021.

[5] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In IEEE International Conference on Computer Vision (ICCV), pages 7063–7072, 2019.

[6] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015.

[7] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 224–236, 2018.

[8] 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, pages 2650–2658, 2015.

[9] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In European conference on computer vision (ECCV), pages 834–849. Springer, 2014.

[10] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. IEEE Transactions on Robotics, 33(2):249–265, 2016.

[11] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In IEEE International Conference on Computer Vision (ICCV), pages 3828–3838, 2019.

[12] X. Han, Y. Tao, Z. Li, R. Cen, and F. Xue. Superpointvo: A lightweight visual odometry based on cnn feature extraction. In 5th International Conference on Automation, Control and Robotics Engineering (CACRE), pages 685–691, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE conference on computer vision and pattern recognition (CVPR), pages 770–778, 2016.

[14] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In International Conference on Pattern Recognition (ICPR), pages 2366–2369, 2010.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[16] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In IEEE International Conference on Robotics and Automation (ICRA), pages 7286–7291, 2018.

[17] Y. Li, Y. Ushiku, and T. Harada. Pose graph optimization for unsupervised monocular visual odometry. In IEEE International Conference on Robotics and Automation (ICRA), pages 5439–5445, 2019.

[18] G. Lu. Image-based localization for self-driving vehicles based on online network adjustment in a dynamic scope. In IEEE International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2022.

[19] G. Lu, L. Nie, S. Sorensen, and C. Kambhamettu. Large-scale tracking for images with few textures. IEEE Transactions on Multimedia, 19(9):2117–2128, 2017.

[20] G. Lu and X.-I. Wong. Taking me to the correct place: Vision-based localization for autonomous vehicles. In IEEE International Conference on Image Processing (ICIP), pages 2966–2970, 2019.

[21] G. Lu, X.-I. Wong, and J. McBride. From mapping to localization: A complete framework to visually estimate position and attitude for autonomous vehicles. In IEEE International Conference on Image Processing (ICIP), pages 3103–3107, 2019.

[22] G. Lu, Y. Yan, N. Sebe, and C. Kambhamettu. Indoor localization via multi-view images and videos. Computer Vision and Image Understanding, 161:145–160, 2017.

[23] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan. Hr-depth: high resolution self-supervised monocular depth estimation. In AAAI Conference on Artificial Intelligence, 2021.

[24] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5667–5675, 2018.

[25] J. Mooser, S. You, U. Neumann, and Q. Wang. Applying robust structure from motion to markerless augmented reality. In 2009 Workshop on Applications of Computer Vision (WACV), pages 1–8, 2009.

[26] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5):1255–1262, 2017.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems (NeurIPS), 32:8026–8037, 2019.

[28] L. Platinsky, M. Szabados, F. Hlasek, R. Hemsley, L. Del Pero, A. Pancik, B. Baum, H. Grimmett, and P. Ondruska. Collaborative augmented reality on smartphones via life-long city-scale maps. In IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pages 533–541, 2020.

[29] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 12240–12249, 2019.

[30] S. Śledź, M. Ewertowski, and J. Piekarczyk. Applications of unmanned aerial vehicle (uav) surveys and structure from motion photogrammetry in glacial and periglacial geomorphology. Geomorphology, page 107620, 2021.

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

[32] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. In Robotics: Science and Systems, 2015.

[33] N. Yang et al. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[34] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[35] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 340–349, 2018.

[36] C. Zhao, Y. Tang, Q. Sun, and A. V. Vasilakos. Deep direct visual odometry. IEEE Transactions on Intelligent Transportation Systems, 2021.

[37] H. Zhou, B. Ummenhofer, and T. Brox. Deeptam: Deep tracking and mapping. In European conference on computer vision (ECCV), pages 822–838, 2018.

[38] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1851–1858, 2017.

[39] Y. Zou, P. Ji, Q. Tran, J. Huang, and M. Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. arXiv:2007.10983, 2020.

[40] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In European Confer-ence on Computer Vision (ECCV), pages 36–53, 2018.