Deep Direct Visual Odometry

Chaoqiang Zhao¹, Yang Tang¹ and Qiyu Sun¹

Abstract—Monocular direct visual odometry (DVO) relies heavily on high-quality images and good initial pose estimation for accuracy tracking process, which means that DVO may fail if the image quality is poor or the initial value is incorrect. In this study, we present a new architecture to overcome the above limitations by embedding deep learning into DVO. A novel self-supervised network architecture for effectively predicting 6-DOF pose is proposed in this paper, and we incorporate the pose prediction into Direct Sparse Odometry (DSO [1]) for robust initialization and tracking process. Furthermore, the attention mechanism is included to select useful features for accurate pose regression. The experiments on the KITTI dataset show that the proposed network achieves an outstanding performance compared with previous self-supervised methods, and the integration with pose network makes the initialization and tracking of DSO more robust and accurate.

1 INTRODUCTION

Simultaneous localization and mapping (SLAM) and visual odometry (VO) supported by monocular [1], [2], stereo [3], [4] or RGB-D [5], [6] cameras, play an important role in various fields, including virtual/augmented reality and autonomous driving. Due to its real-time performance and low computational complexity, VO has attracted more and more attention in robotic pose estimation [7]. In recent years, different kinds of approaches have been proposed to solve VO problems, including direct methods [1], semi-direct methods [2] and feature-based methods [6].

Simultaneously recovering ego-motion and 3D scene geometry is a fundamental topic. Feature-based methods dominated this field for a long time. During tracking, the key-points on the new frame are extracted, and their descriptors like ORB are calculated to find the 2D-2D or 3D-2D correspondences [8]. The robustness of feature-based methods depends on the accuracy of feature matching, which makes it difficult to work in low-textured and repetitive textured contexts [2].

In contrast to feature-based methods, semi-direct and direct methods use the photometry information directly and eliminate the need to calculate and match feature descriptors. Although direct methods have shown to be more robust in the case of motion

¹The authors are with the Key Laboratory of Advanced Control and Optimization for Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China. E-mail: tangtany@gmail.com; yangtang@ecust.edu.cn

(a) DDSO on Seq. 10

(b) DSO on Seq. 10

Fig. 1. The example trajectories of our model (DDSO) and DSO running on KITTI odometry sequence 10 show that our proposed model are more robust than DSO in initialization. (a). Our model (DDSO) can successfully complete the initialization, even if there are large motions and visual field changes (turning) in the initial stage. (b). DSO can smoothly finish the initialization process only when there is a small view change (straight road).
blur or high repetitive textured scenes, this method is sensitive to the photometric changes, which means that a photometric camera model should be considered for better performance [1], [9]. Furthermore, the pose solution of direct methods depends on the image alignment algorithm, which heavily relies on the initial value provided by a constant motion model. Therefore, direct methods are easy to fail if the image quality is poor or the initial pose estimation is incorrect. During initialization process, the constant motion model is not applicable due to the lack of prior motion information in the initialization stage. As a result, the initial pose is initialized as a unit matrix, which is inaccurate and will lead to the failure of the initialization. Hence, the accurate initialization and tracking in direct methods require a fairly good initial estimation as well as high-quality images. If the pose of camera has a great change or the camera is in a high dynamic range (HDR) environment, the direct methods are difficult to finish initialization and accurate tracking.

Compared with the traditional VO methods, deep learning models do not rely on high-precision features correspondence or high-quality images [10]. Because of their ability of high-level features extraction, deep learning-based methods have been widely used in image processing and made considerable progress. Recently, the deep models for VO problems have been proposed by trained via ground truth [11]–[13] or jointly trained with other networks in an self-supervised way [14]–[16]. They use the loss function to help the neural network learn internal geometric relations. Due to the lack of local or global consistency optimization, the accumulation of errors and scale drift prevent the pure deep VO from being used directly. In particular, the 3D scenes geometry cannot be visualized because there is no mapping thread, which makes subsequent navigation and obstacle avoidance impossible.

Considering the advantages of deep learning in high-level features extraction and the robustness in HDR environments, we incorporate deep learning into DSO, called deep direct sparse odometry (DDSO). The local consistency optimization of pose estimation obtained by deep learning is carried out by the traditional direct method. Meanwhile, 3D scene geometry can be visualized with the mapping thread of DSO. Most importantly, DSO are capable of obtain more robust initialization and accurate tracking with the aid of deep learning.

The main contributions are listed as follows:

- An efficient pose prediction network (PoseNet) is designed for pose estimation and trained in a self-supervised manner. Meanwhile, a soft-attention model and STM module are used to improve the feature manipulation ability of our model.
- A new direct VO framework cooperated with PoseNet is proposed to improve the initialization and tracking process. To the best of our knowledge, this is the first time to apply the pose network to the traditional direct methods.
- Both the PoseNet and DDSO framework proposed in this paper show outstanding experimental results on KITTI dataset [17]. Compared with previous works, our PoseNet is simpler and more effective. Our DDSO also achieves more robust initialization and accurate tracking than DSO.

The organization of this work is as follows: In section II, the related works on monocular VO are discussed. Section III introduces our self-supervised PoseNet framework and DDSO model in detail. Section IV shows the experimental results of our PoseNet and DDSO on KITTI. Finally, this study is concluded in section V.

2 RELATED WORK

2.1 Geometric-based Architecture

The traditional sparse feature-based method [9] is used to estimate the transformation from a set of keypoints by minimizing the reprojection error. Because of suffering from the heavy cost of feature extraction and matching, this method has a low speed and poor robustness in low-texture scenes. Engel et al. [18] present a semi-dense direct framework that employs photometric errors as a geometric constraint to estimate the motion. However, this method optimizes the structure and motion in real-time, and tracks all pixels with gradients in the frame, which is computationally expensive. Therefore, a direct and sparse method is then proposed in [1], which has been manifested more accurate than [18], by optimizing the poses, camera intrinsics and geometry parameters based on a nonlinear optimization
Fig. 2. Our self-supervised network architecture. (a) In order to achieve a better pose prediction, we use 7 convolution layers with kernel size 3 for feature extraction, the full connected layers and attention model. (b) A soft-attention model is used for feature association and selection. The reweighted features are used to predict 6-DOF relative pose. (c) A STM model is used to replace the common skip connection between encoder and decoder and selective transfer characteristics in DepthNet. (d) The single-frame DepthNet adopts the encoder-decoder framework with a selective transfer model, and the kernel size is 3 for all convolution and deconvolution layers. See section III-A for more details.

framework. Then, the studies in [19]–[21] are used to solve the scale ambiguity and scale drift of [1]. An approach with a higher speed that combines the advantage of feature-based and direct methods is designed by Forster et al. [2]. However, these approaches in [1], [2] are sensitive to photometric changes and rely heavily on accurate initial pose estimation, which make initialization difficult and easy to fail in the case of large motion or photometric changes. Recently, the methods based on deep learning are also employed to recover scale [22], improve the tracking [23] and mapping [24]. In this paper, we leverage the proposed pose network into DSO to improve the robustness and accuracy of the initialization and tracking.

2.2 Deep learning-based Architecture

With the development of deep neural networks, end-to-end pose estimation has achieved great progress. Alex et al. [12] train a convolution neural network (CNN) to predict the position of camera in a supervised manner, and this method shows some potentials in camera localization. Wang et al. [11] consider the constraint of temporal information by using a recurrent model to estimate 6-DOF transformation, while training the proposed networks require large quantities and an expensive cost for ground truth data. In recent years, self-supervised-based approaches have become more popular because of freeing from groundtruth. Many effective frameworks for predicting motion have been proposed in [14], [25], and the pose network is trained jointly with a depth network by a view reconstruction loss. Moreover, the optical flow network is also added to the above framework [15], [16], [26], so that the performance of the pose network is improved by additional optical flow constraints. In order to eliminate the influence of dynamic factors and occlusions on the network, the motion segmentation network [16] or mask model [27] is trained together with the pose network for a better result. Furthermore, the inertial information is also used in pose prediction, resulting in a promising performance [13], [28]. Different from the above methods, the deep learning-based VO framework proposed in this paper does not need to have additional complex modules like FlowNet [15], [16] and MaskNet [16], inertial information or loss function constraints, but achieving competitive performance with the inclusion of the attention mechanism.

3 METHODS

In this section, we introduce the architecture of our deep self-supervised neural networks for pose
estimation in part A and describe our deep direct sparse odometry architecture (DDSO) in part B.

### 3.1 Self-supervised network

Instead of using the expensive ground truth for training the PoseNet, a general self-supervised framework is considered to effectively train our network in this study (as shown in Fig. 3). The PoseNet is trained by the RGB sequences composed of a target frame \( I_t \) and its adjacent frame \( I_{t-1} \) and regresses the 6-DOF transformation \( \hat{T}_{t,t-1} \) of them. Simultaneously, a depth map \( \hat{D}_t \) of the target frame is generated by the DepthNet. The geometry constraints between the two model outputs serve as a training monitor that help the model learn the geometric relations between adjacent frames. The key supervisory signal for our models comes from the view reconstruction loss \( \mathcal{L}_c \) and smoothness loss \( \mathcal{L}_{\text{smooth}} \):

\[
\mathcal{L} = \sum L_{\mathcal{C}}^s + \alpha \sum L_{\text{smooth}}^s, \tag{1}
\]

where \( \alpha \) is a smoothness loss weight, \( s \) represents pyramid image scales. The structure of overall function is similar to [14], but the loss terms are calculated differently and described in the following.

**View construction as supervision:** During training, two consecutive frames including target frame \( I_t \) and source frame \( I_{t-1} \) are concatenated along channel dimension and fed into PoseNet to regress 6-DOF camera pose \( \hat{T}_{t,t-1} \). Our DepthNet takes a single target frame \( I_t \) as input and output the depth prediction \( \hat{D}_t \) for per-pixel. As indicated in Eq. (2), we can get the pixel correspondence of two frames by geometric projection based rendering module [29]:

\[
p_{t-1} \sim K \hat{D}_t(p_t) K^{-1} p_t, \tag{2}
\]

where \( K \) is the camera intrinsics matrix. Notice that \( p_t \) is continuous on the image while the projection is discrete. In order to warp the source frame \( I_{t-1} \) to target frame \( I_t \) and get a continuous smooth reconstruction frame \( \hat{I}_{t-1} \), we use the differentiable bilinear interpolation mechanism. We assume that the scenes used in training are static and adopt a robust image similarity loss [30] for the consistence of the two views \( I_t, \hat{I}_{t-1} \):

\[
\mathcal{L}_c = \beta \frac{1 - \text{SSIM}(I_t, \hat{I}_{t-1})}{2} - (1 - \beta) ||I_t - \hat{I}_{t-1}||_1, \tag{3}
\]

where \( \text{SSIM}(I_t, \hat{I}_{t-1}) \) stands for the structural similarity [31] between \( I_t \) and \( \hat{I}_{t-1} \).

**Smoothness constraint of depth map:** This loss term is used to promote the representation of geometric details. There are many planes in the scenes, and the depth of adjacent pixels in the same plane presents gradient changes. Therefore, this paper adopts the second derivative of the same plane depth to promote depth smoothness, which is different from [15]. Hence, the improved smoothness loss \( \mathcal{L}_{\text{smooth}} \) is expressed as:

\[
\mathcal{L}_{\text{smooth}} = \sum_{p_t} |\nabla(\nabla D(p_t))| \cdot (e^{-|\nabla p_t|})^T, \tag{4}
\]

where \( |\cdot| \) represents absolute value, \( \nabla \) stands for the vector differential operator, and \( T \) refers to the transpose operation.

![Fig. 3. The self-supervised training framework in our paper.](image)

Our self-supervised network architecture is inspired by Zhou et al.’s work [14] while making several improvements (as shown in Fig. 2). Our PoseNet follows the basic structure of FlowNetS [32] because of its more effective feature extraction manner. We use 7 CNN layers for high-level feature extraction and 3 full-connected layers for a better pose regression. The main difference between our PoseNet and the previous works [15], [16] is the use of attention mechanisms. A soft-attention model is designed in PoseNet to reweight the extracted features. Meanwhile, a selective transfer model (STM) [33] with the ability to selectively deliver characteristic information is also added into the depth network to replace the skip connection. Since the training of DepthNet and PoseNet is coupled, the
Fig. 4. The DDSO pipeline. Our work augments the DSO framework with the pose prediction module (PoseNet). Every new frame is fed into the proposed PoseNet with last frame to regress a relative pose estimation. The predicted pose is used to improve initialization and tracking in DSO.

improvement of DepthNet can improve the performance of PoseNet indirectly.

**Soft-attention model:** Similar to the widely applied self-attention mechanism [28], [34], we use a soft-attention model in our pose network for selectively and deterministically models feature selection. This function reweights the feature $a_e$ extracted by the encoder, which allows the relatively important features to be selected and highlighted. Then, the reweighted feature vector $a_v$ is used for pose regression:

$$T_{ij} = f_{\text{sigmoid}}(a_v) = f_{\text{sigmoid}}(a_e \circ f_{\text{sigmoid}}(a_e)).$$ (5)

where $f_{\text{sigmoid}}$ is a full connection layer with sigmoid function, $\circ$ denotes the dot product of the vector.

**Selective Transfer model:** Inspired by [33], a selective model STM is used in depth network. The encoder feature $f_{\text{enc}}^l$ of $l$-th layer is sent to STM, and selected by the hidden state $s^{l+1}$ from the $l+1$-th layer:

$$s^{l+1} = W_t * D_{\text{deconv}}(s^{l+1}),$$

$$r^l = \sigma(W_r * [f_{\text{enc}}^l, s_{l+1}]),$$

$$z^l = \sigma(W_z * [f_{\text{enc}}^l, s^{l+1}]),$$

$$s^l = r^l \cdot s^{l+1},$$

$$f^l_i = (1 - z^l) \cdot s^{l+1} + z^l \cdot \tanh(W_h * [f_{\text{enc}}^l, s^l]),$$

where $D_{\text{deconv}}(\cdot)$ stands for deconvolution while $W_\cdot *$ refers to different layers of convolution. $[\cdot, \cdot]$ means the concatenation step. $\cdot$ stands for multiply, and $\sigma(\cdot)$ is the sigmoid function.

3.2 Deep Sparse Visual Odometry

In this paper, our deep direct sparse odometry (DDSO) can be regarded as the cooperation of PoseNet and DSO. Firstly, the overall framework of DDSO is discussed briefly. DSO is a keyframe-based approach, where 5-7 keyframes are maintained in the sliding window and their parameters are jointly optimized by minimizing photometric errors in the current window. New frames are tracked with respect to the nearest keyframe using a multiscale image pyramid, a two-frame image alignment algorithm and an initial transformation. When a new frame is captured by camera, all active points in the sliding window are projected into this frame (Eq. (7)), resulting in a photometric error $E_{pj}$ (Eq. (8)). Then the total photometric error $E_{\text{total}}$ (Eq. (9)) of the sliding window is optimized by the Gauss-Newton algorithm and used to calculate the relative transformation $T_{ij}$.

$$p' = \pi_c(R_{\pi^{-1}_c}(p, d_p) + t), \quad \begin{bmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{bmatrix} = T_{ij},$$ (7)

$$E_{pj} := \sum_{p \in \mathcal{N}_p} w_p \| I_j[p'] - I_i[p] \|_\gamma,$$ (8)

$$E_{\text{total}} := \sum_{i \in \mathcal{F}} \sum_{p \in \mathcal{P}_i} \sum_{j \in \text{obs}(p)} E_{pj},$$ (9)

where $\pi_c$ is the projection function: $\mathbb{R}^3 \rightarrow \omega$ while $\pi^{-1}_c$ is back-projection. $p'$ stands for the projected point position of $p$ with inverse depth $d_p$. $T_{ij}$ is the transformation between two related frames $I_i$ and $I_j$. $\mathcal{F}$ is a collection of frames in the sliding window, and $\mathcal{P}_i$ refers to the points in frame $i$. $\text{obs}(p)$ means that the points are visible in the current frame.
Since the whole process can be regarded as a nonlinear optimization problem, an initial transformation should be given and iteratively optimized by the Gauss-Newton method. Therefore, the initial transformation especially orientation is very important for the whole tracking process. During tracking, a constant motion model is applied for initializing the relative transformation between the current frame and last key-frame in DSO, as shown in Eq. (10) and Eq. (11), assuming that the motion \( T_{t,t-1} \) between the current frame \( I_t \) and last frame \( I_{t-1} \) is the same as the previous one \( T_{t-1,t-2} \):

\[
T_{t,t-1} = T_{t-1,t-2} = T_{t-1,w} \cdot T_{t-2,w}^{-1},
\]

\[
T_{t,kf} = T_{t,t-1} \cdot T_{t-1,kf} = T_{t,kf} \cdot T_{t-1,w} \cdot T_{kf,w}^{-1},
\]

where \( T_{t-1,w}, T_{t-2,w}, T_{kf,w} \) are the poses of \( I_{t-1}, I_{t-2}, I_{kf} \) in world coordinate system.

Considering that it is not reliable to use only the initial transformation provided by the constant motion model, DSO attempts to recover the tracking process by initializing the other 3 motion models and 27 different small rotations when the image alignment algorithm fails, which is complex and time consuming. Since there is no motion information as a priori during initialization process, the transformation is initialized to the identity matrix, and the inverse depth of the point is initialized to 1.0. In this process, the initial value of optimization is meaningless, resulting in inaccurate results and even initialization failure.

For this reason, we utilize a PoseNet to provide an accurate initial transformation especially orientation for initialization and tracking process in this paper. With the help of PoseNet, a better pose estimation can be regarded as a better guide for initialization and tracking. As shown in Fig. 4, deep direct sparse odometry (DDSO) builds on the monocular DSO without photometric camera calibration, and the pose predictions provided by our PoseNet are used to improve DSO in both initialization and tracking process. We replace the initial pose conjecture generated by the constant motion model with the output of PoseNet, incorporating it into the two-frame direct image alignment algorithm. When a new frame comes, a relative transformation \( T_{t,t-1} \) is regressed by PoseNet from the current frame \( I_t \) and last frame \( I_{t-1} \), which is regarded as the initial value of the image alignment algorithm. Due to a more accurate initial value provided for the nonlinear optimization process, the robustness of DSO tracking is improved.

### 4 Experiment

We evaluate our PoseNet as well as DDSO against the state-of-the-art methods on the publicly available KITTI dataset [17].

#### 4.1 Pose estimation

**Training detail:** We implement the architecture with Tensorflow framework [35] and train on a NVIDIA RTX 2080Ti GPU. Our parameter settings are similar to previous works [14], [15]. The loss weights are set to be \( \alpha = 0.5 \) and \( \beta = 0.85 \). Our network is trained by ADAM optimizer with \( \beta_1 = 0.9 \). During training, the resolution of input images is adjusted to 128 \( \times \) 416. Both the batch normalization and ReLUs are used for all layers except for the output layer. The learning rate is initialized as 0.0002 and the mini-batch is set as 4. The training converges after about 200K iterations.

**Evaluation:** We have evaluated the performance of our PoseNet on the KITTI VO sequence. We use 00-08 sequences of the KITTI odometry for training and 09-10 sequences for evaluating. Our PoseNet can flexibly set the number of input frames and last key-frame in DSO, as shown in Eq. (10) and Eq. (11), assuming that the motion \( T_{t,t-1} \) between the current frame \( I_t \) and last frame \( I_{t-1} \) is the same as the previous one \( T_{t-1,t-2} \):

| Method              | Frame\(^1\) | \( \text{NP}\)\(^2\) | Seq. 09\(^3\) | Seq. 10\(^4\) |
|--------------------|-------------|----------------|--------------|--------------|
| ORB-SLAM (full)    | –           | –              | 0.014 ± 0.008 | 0.012 ± 0.011 |
| SfM Learner [14]   | 5           | 33.2M           | 0.021 ± 0.017 | 0.020 ± 0.015 |
| Vid2Depth [25]     | 5           | 33.2M           | 0.013 ± 0.010 | 0.012 ± 0.011 |
| Geonet [15]        | 5           | 58.5M           | 0.012 ± 0.007 | 0.012 ± 0.009 |
| CC [16]            | 5           | 127.6M          | 0.012 ± 0.007 | 0.012 ± 0.008 |
| PoseNet (ours\(^5\)) | 5           | 35.0M           | 0.014 ± 0.007 | 0.013 ± 0.010 |
| PoseNet (ours\(^5\)) | 5           | 36.4M           | 0.013 ± 0.007 | 0.012 ± 0.009 |
| Wang et al. [27]   | 3           | –              | 0.009 ± 0.005 | 0.008 ± 0.007 |
| PoseNet (ours\(^5\)) | 3           | 36.4M           | 0.008 ± 0.005 | 0.008 ± 0.007 |
| SfM Learner\(^\ast\) [14] | 2           | 33.2M           | 0.0103 ± 0.0096 | 0.0097 ± 0.0111 |
| Geonet \(^\ast\) [15] | 2           | 58.5M           | 0.0057 ± 0.0038 | 0.0058 ± 0.0056 |
| PoseNet\(^\ast\) (ours\(^5\)) | 2           | 36.4M           | 0.0057 ± 0.0036 | 0.0056 ± 0.0057 |

1 - The length of trajectories used for evaluation.
2 - Number of parameters in the network, ‘M’ denotes million.
3 - Absolute Trajectory Error (ATE) on KITTI sequence 09 and 10.
4 - Our PoseNet is trained without attention and STM modules.
5 - Evaluation of pose prediction between adjacent frames. We download, process and evaluate the results they publish.
4.2 Deep Direct Sparse Odometry

For DDSO, we compare its initialization process as well as tracking accuracy on the odometry sequences of KITTI dataset against the state-of-the-art direct methods, DSO (without photometric camera calibration). The python package, evo [36], is used to evaluate the trajectory errors of DDSO and DSO. We use the KITTI odometry 00-06 sequences for retraining our PoseNet with 3-frame input and 07-10 sequences for testing on DSO and DDSO. Fig. 5 shows the estimated trajectories (a)-(d) on sequences 07-10 drawn by evo [36]. It verifies that our framework works well, and the strategy of replacing pose initialization models including a constant motion model with pose network is effective and even better. Then, both the absolute pose error (APE) and relative pose error (RPE) of trajectories generated by DDSO and DSO are computed by the trajectory evaluation tools in evo. As shown in Table 2, DDSO achieves better performance than

and 5-frame trajectories predicted by our PoseNet and compare with the previous state-of-the-art self-supervised works [14]-[16], [25], [27]. As shown in Table 1, our method achieves better result than “ORB-SLAM (full)” and better performance in 3-frame and adjacent frames pose estimation. For 5-frame trajectories evaluation, the state-of-the-art method CC [16] needs to train 3 parts iteratively, while we only need train 1 part once for 200K iterations. Hence, the simple network structure makes our training process more convenient. Although there are no additional complex networks (FlowNet [15], MaskNet [14], SegmentationNet [16]) or additional loss function constraints (ICP Loss [25], Collaboration Loss [16], Geometric Consistency Loss [15]) in our model, decent performance is achieved. Compared with our PoseNet without attention and STM module, the result of our full PoseNet shows the effectiveness of our soft-attention and STM modules.

| Sequence | Absolute Pose Error$^2$ (APE) | Relative Pose Error$^2$ (RPE) |
|----------|--------------------------------|--------------------------------|
|          | Max Mean Min MSE Std           | Max Mean Min MSE Std           |
| seq 07 DSO | 76.35 33.10 1.19 38.22 19.10 | 2.68 0.20 0.0004 0.287 0.215 |
| Ours     | 44.04 13.81 0.47 16.70 9.40  | 0.32 0.11 0.0007 0.139 0.081 |
| seq 08 DSO | 287.44 74.89 10.62 92.84 54.865 | 1.44 0.386 0.002 0.473 0.273 |
| Ours     | 288.33 74.47 10.5 92.69 55.19 | 1.43 0.385 0.002 0.472 0.273 |
| seq 09 DSO | 184.96 57.57 2.83 69.16 38.33 | 3.168 0.302 0.002 0.418 0.288 |
| Ours     | 132.94 55.02 1.57 64.68 36.61 | 1.168 0.234 0.002 0.261 0.115 |
| seq 10 DSO | 135.66 55.35 10.86 44.12 26.14 | 4.006 0.286 0.0028 0.4213 0.3089 |
| Ours     | 66.88 13.29 2.82 21.92 12.59 | 0.73 0.15 0.0018 0.233 0.1785 |

ICD$^1$ means whether the initialization can be completed within the first 20 frames

$^2$ Lower is better

---

Fig. 5. Sample trajectories comparing the DDSO with DSO, and the ground truth in metric scale. DDSO shows a better odometry estimation in terms of both initialization and tracking.
DSO on the sequences 07-10. One reason is that the good initialization improves the tracking process, and the other is that the transformation computed by the constant motion model is replaced by the one produced by PoseNet during tracking. Table 2 also shows the advantage of DDSO in initialization on sequence 07-10. Because of its inability of handling several brightness changes and its initialization process, DSO cannot complete the initialization smoothly and quickly on sequence 07, 09 and 10. However, the photometric has little effect on the pose network, and the nonsensical initialization is replaced by the relatively accurate pose estimation regressed by PoseNet during initialization, so that DDSO can finish the initialization successfully and stably. Therefore, with the help of PoseNet, our DDSO achieves robust initialization and more accurate tracking than DSO. What’s more, the cooperation with traditional methods also provides a direction for the practical application of the current learning-based pose estimation.

5 CONCLUSION

In summary, we present a novel monocular direct VO framework DDSO, which incorporates the PoseNet proposed in this paper into DSO. The initialization and tracking are improved by using the PoseNet output as an initial value into image alignment algorithm. For PoseNet, it is designed with an attention mechanism and trained in a self-supervised manner by the improved smoothness loss and SSIM loss, achieving an decent performance against the previous self-supervised methods. Our evaluation conducted on the KITTI odometry dataset demonstrates that DDSO outperforms the state-of-the-art DSO by a large margin. Meanwhile, the initialization and tracking of our DDSO are more robust than DSO. The key benefit of our DDSO framework is that it allows us to obtain robust and accuracy direct odometry without photometric calibration [9]. What’s more, since the initial pose including orientation provided by the pose network is more accurate than that provided by the constant motion model, this idea can also be used in the other methods which solve poses by image alignment algorithms. Nevertheless, there are still shortcomings that need to be addressed in the future. The scale drift still exists in our proposed method, and we plan to integrate inertial information and proper constrains into the estimation network to improve the scale drift.

REFERENCES

[1] 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.
[2] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” IEEE Transactions on Robotics, vol. 33, no. 2, pp. 249–265, 2016.
[3] J. Mo and J. Sattar, “DSVO: Direct Stereo Visual Odometry,” arXiv preprint arXiv:1810.03963, 2018.
[4] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008, pp. 3946–3952.
[5] T. Schops, T. Sattler, and M. Pollefeys, “BAD SLAM: Bundle Adjusted Direct RGB-D SLAM,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 134–144.
[6] 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, vol. 33, no. 5, pp. 1255–1262, 2017.
[7] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” IEEE robotics & automation magazine, vol. 18, no. 4, pp. 80–92, 2011.
[8] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in ICCV, vol. 11, no. 1. Citeseer, 2011, p. 2.
[9] P. Bergmann, R. Wang, and D. Cremers, “Online photometric calibration of auto exposure video for realtime visual odometry and slam,” IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 627–634, 2017.
[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[11] 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.
[12] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-dof camera relocalization,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.
[13] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem,” in Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3995–4001.
[14] 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.
[15] 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.
[16] 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 Proceedings of the IEEE Conference on
[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.

[18] 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.

[19] R. Wang, M. Schworer, and D. Cremers, “Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2198–2204.

[20] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 817–833.

[21] Y. Zou, Z. Luo, and J.-B. Huang, “Df-net: Unsupervised joint learning of depth and flow using cross-task consistency,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 36–53.

[22] G. Wang, H. Wang, Y. Liu, and W. Chen, “Unsupervised Learning of Monocular Depth and Ego-Motion Using Multiple Masks,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 4724–4730.

[23] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, “Selective Sensor Fusion for Neural Visual-Inertial Odometry,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10 542–10 551.

[24] C. Fehn, “Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” in Stereoscopic Displays and Virtual Reality Systems XI, vol. 5291. International Society for Optics and Photonics, 2004, pp. 93–104.

[25] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270–279.

[26] W. Zhou, B. Alan Conrad, S. Hamid Rahim, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Process, vol. 13, no. 4, pp. 600–612, 2004.