2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

FlowNorm: A Learning-based Method for Increasing Convergence Range of Direct Alignment

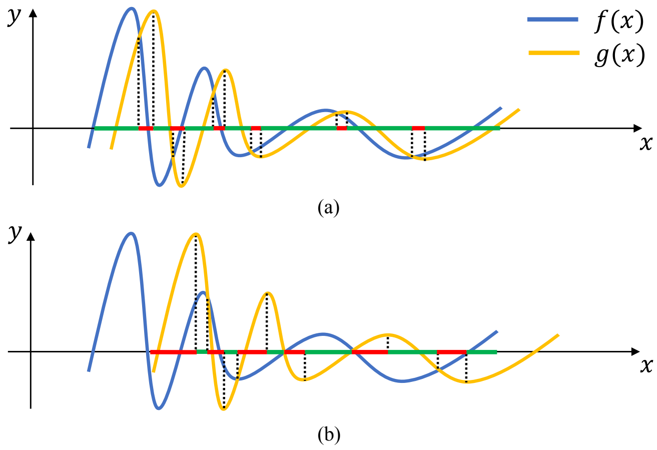
Ke Wang, Kaixuan Wang, and Shaojie Shen

Abstract— **Many approaches have been proposed to estimate camera poses by directly minimizing photometric error. However, due to the non-convex property of direct alignment, proper initialization is still required for these methods. Many robust norms (e.g. Huber norm) have been proposed to deal with the outlier terms caused by incorrect initializations. These robust norms are solely defined on the magnitude of each error term. In this paper, we propose a novel robust norm, named FlowNorm, that exploits the information from both the local error term and the global image registration information. While the local information is defined on patch alignments, the global information is estimated using a learning-based network. Using both the local and global information, we achieve a large convergence range in which images can be aligned given large view angle changes or small overlaps. We further demonstrate the usability of the proposed robust norm by integrating it into the direct methods DSO and BA-Net, and generate more robust and accurate results in real-time.**

I. Introduction

Direct methods are widely used to solve visual odometry and monocular stereo problems [1]–[4]. By directly minimizing the photometric error between pixels in the source frame and the target frame, camera poses and scene geometry can be estimated in the joint optimization process. Compared with indirect methods [5]–[8], which solve the problem by minimizing the reprojection error between matched sparse features, direct methods avoid the pre-processed feature matching step and can utilize more pixels in the image. However, intensity-based optimization is prone to local minima due to the non-convex property of complex images.

Recent years, many approaches have been proposed to expand the convergence range of direct methods. SVO [9] combines matched feature points with photometric optimization. However, although matched features can provide pose initialization for further optimization, they rely on textures of the environment and are prone to outliers. With the help of learning-based methods, many researchers have proposed networks [10], [11] to generate smooth feature maps for direct optimization. Compared with the image intensity domain, optimization on feature maps shows advantages in convergence ranges. For example, BA-Net [10] can estimate camera poses given images with small overlaps. LS-Net [12] uses an end-to-end trained network as a solver for two-frame monocular stereo problems. Learning-based methods achieve superior performance on evaluation datasets, such as RGB-D datasets or the KITTI dataset, but have not been widely used on robotic platforms. The reason may be the limited computation resources of general robotic platforms and the diversity of robotic application scenes.



(a)

(b)

Fig. 1. A simple example to show the different contributions of points in aligning two functions: $\mathrm{argmin}_t \sum_x \|g(x+t)-f(x)\|_2$. Point x with $(g(x+t)-f(x))g'(x+t) < 0$ contributes to the optimization of t and is marked in green, while x with $(g(x+t)-f(x))g'(x+t) > 0$ counteracts the optimization and is marked in red. As shown in (a), with good initialization, most of the points are positive to the optimization. However, with worse initialization, negative points make the optimization fall into local minima.

One of the contributions of the paper is a study of the direct optimization process followed by the design of a robust norm for the optimization. Due to the problem of non-convexity, during the photometric minimization (or feature consistency minimization in learning-based methods), not all pixels contribute to the convergence. The difference in pixels depends on both the local texture and the global pose initialization which establishes pixels correspondences. We illustrate the convergence problem in Fig. 1. As shown, for good initializations, most of the correspondences contribute to the final estimation. However, given a bad initialization, most of the correspondences will suppress the convergence. Poor correspondences make the optimization fall into local minima. Based on this observation, we propose the flow norm that combines a low-accuracy optical flow prediction network to distinguish which correspondences will suppress the convergence of direct alignment.

In summary, the contributions of our paper are the following:

- We propose a new norm to expand the convergence range of the traditional nonlinear solver for the direct alignment problem.
- To the best of our knowledge, the proposed method

The authers are with Department of Electronic and Computer Engineering , Hong Kong University of Science and Technology, kwangbd@connect.ust.hk

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

is the first that can distinguish which correspondence will suppress solver convergence in the direct alignment situation.
- We build FlowNorm versions of DSO and BA-Net, and the FlowNorm DSO retains the real-time property.

To demonstrate the effectiveness of our method, we evaluate it on the SceneNN dataset [13], TUM-MonoVO dataset [14] and ICL-NUIM dataset [15], showing that FlowNorm versions consistently outperform the original versions.

II. Related Work

Semi-dense visual odometry [4] is a pioneering work that tracks a monocular camera in real-time using direct alignment algorithm. SVO [9] uses matched features to calculate an initialization pose for joint optimization. Following the idea of the direct method, Engel proposed LSD-SLAM [3] ,which solves the camera pose using keyframes with depth values. DSO [16] is the baseline direct alignment work, which jointly optimizes all model parameters, including geometry represented as inverse depth and camera motion. DSO further integrates a full photometric calibration, accounting for exposure time, lens vignetting, and non-linear response functions. Although all these methods feature real-time efficiency and high accuracy, they rely on incrementally tracking the camera poses to ensure large overlaps and proper initialization.

To increase the convergence range of the direct methods, many learning-based methods have been proposed to replace the intensity map with feature maps. BA-Net [10] formulates Bundle Adjustment (BA) as a differentiable layer and utilizes a standard encoder-decoder network to generate the feature map and depth map. Camera poses and depth maps are optimized by minimizing the feature consistency between projected pixels. Benefiting from the generated feature maps, BA-Net expands the convergence range of direct alignment. GN-Net [11] uses a novel Gauss-Newton loss for training deep feature maps. The direct alignment in GN-Net, based on minimizing the feature metric error, achieves robust performance under dynamic lighting or weather changes. These two approaches nicely combine traditional direct alignment and deep learning techniques.

LS-Net [12] uses an end-to-end trained network to replace the traditional nonlinear solver. Given a photometric error map and a Jacobian matrix, LS-Net estimates the updated depth map and camera motion. Although it achieves impressive results on datasets, the generalization ability of LS-Net has not been demonstrated.

In this paper, we propose a different solution that improves the robustness of direct optimization. The core of the contribution is a robust norm that distinguishes error terms using both local and global information. Different from most of the learning-based methods that use a heavy network to generate high-dimensional feature maps, we utilize a lightweight network to improve both the robustness and accuracy of the state-of-the-art methods, with an overhead of only $14\ ms$.

III. Direct Alignment Revisited

Before introducing our enhanced direct alignment algorithm, we revisit the classic direct alignment to give a better understanding of where difficulties lie, and why our method is desirable. We only introduce the most relevant content, and refer the readers to [16] for a more comprehensive introduction.

Given a target/source image pair I_s and I_t, the direct alignment problem is formulated as estimating the relative transformation $\boldsymbol{T}$ between the image pair, and $d_i \in D = \{d_i | i = 1 \cdots N\}$, which are the depths of the pixels $\boldsymbol{p_{si}} \in P = \{\boldsymbol{p_{si}} | i = 1 \cdots N\}$ at the image I_s. Let $\boldsymbol{\mathcal{X}} = \{\boldsymbol{T}, \boldsymbol{D}\}$ and we can estimate $\boldsymbol{\mathcal{X}}$ by minimizing the norm of the photometric error

$$\hat{\boldsymbol{\mathcal{X}}} = \underset{\boldsymbol{\mathcal{X}}}{\arg\min} \sum_{i=1}^{N} |e_i(\boldsymbol{\mathcal{X}})| , \tag{1}$$

where $|\cdot|$ donates the L1 norm or Huber norm of a vector, N is the number of selected pixels, and the photometric error

$$e_i(\boldsymbol{\mathcal{X}}) = I_t(\boldsymbol{p'_{ti}}) - I_s(\boldsymbol{p_{si}}) \tag{2}$$

measures the intensity difference between the ith pixel $\boldsymbol{p_{si}}$ at I_s and its corresponding pixel $\boldsymbol{p'_{ti}}$ at I_t. $\boldsymbol{p'_{ti}}$ is computed by the projection function

$$\boldsymbol{p'_{ti}} = \pi(\boldsymbol{p_{si}}, \boldsymbol{T}, d_i) = s\boldsymbol{K}\boldsymbol{T}d_i\boldsymbol{K^{-1}}\boldsymbol{p_{si}}, \tag{3}$$

which projects 2D point $\boldsymbol{p_{si}}$ from I_s to I_t, where d_i is the depth value of $\boldsymbol{p_{si}}$ at I_s, $\boldsymbol{K}$ and s are the camera's intrinsic matrix and a scale factor respectively.

The general strategy to minimize Eq. (1) is the Gaussian-Newton (GN) or Levenberg-Marquardt (LM) algorithms [17]. The GN and LM methods are both iterative methods. At the jth iteration, the GN algorithm solves for an optimal update

$$\Delta\boldsymbol{\mathcal{X}_j} = -(\boldsymbol{J_j}^{\boldsymbol{T}}\boldsymbol{J_j})^{-1}\boldsymbol{J_j}^{\boldsymbol{T}}\boldsymbol{E_j}. \tag{4}$$

Here $\boldsymbol{E_j} = [e_1(\boldsymbol{\mathcal{X}_j}), e_2(\boldsymbol{\mathcal{X}_j}), \cdots, e_N(\boldsymbol{\mathcal{X}_j})]$, where $\boldsymbol{\mathcal{X}_j}$ is the initial parameters at the jth iteration. Let $\boldsymbol{\delta}$ denotes a small $\boldsymbol{se(3)}$ perturbation around $\boldsymbol{\mathcal{X}_j}$, $\boldsymbol{J_j}$ is the Jacobian matrix of $\boldsymbol{E_j}$ with respect to $\boldsymbol{\delta}$. Let $\boldsymbol{p'_{ti}}$ represent the projection position of $\boldsymbol{p_{si}}$ at I_t based on the parameters $\boldsymbol{\mathcal{X}_j}$. The ith row of $\boldsymbol{J_j}$ is

$$\boldsymbol{J_j(i)} = \left[\frac{\partial e_i(\boldsymbol{\mathcal{X}_j})}{\partial I_t(\boldsymbol{p'_{ti}})} \frac{\partial I_t(\boldsymbol{p'_{ti}})}{\partial \boldsymbol{p'_{ti}}} \frac{\partial \boldsymbol{p'_{ti}}}{\partial \boldsymbol{\delta}} \right], \tag{5}$$

where $\frac{\partial e_i(\boldsymbol{\mathcal{X}_j})}{\partial I_t(\boldsymbol{p'_{ti}})}$ and $\frac{\partial \boldsymbol{p'_{ti}}}{\partial \boldsymbol{\delta}}$ are smooth compared with the increment $\Delta\boldsymbol{\mathcal{X}_j}$. In contrast, $\frac{\partial I_t(\boldsymbol{p'_{ti}})}{\partial \boldsymbol{p'_{ti}}}$ is much less smooth. As found in DSO, $\frac{\partial I_t(\boldsymbol{p'_{ti}})}{\partial \boldsymbol{p'_{ti}}}$ is only valid in a 1-2 pixel radius. Hence the effective optimization requires that all parameters involved in computing $\boldsymbol{p'_{ti}}$ should be initialized sufficiently accurately to be off by no more than 1-2 pixels. However, giving accurate initialization is difficulty when there is a large view change between I_s and I_t.

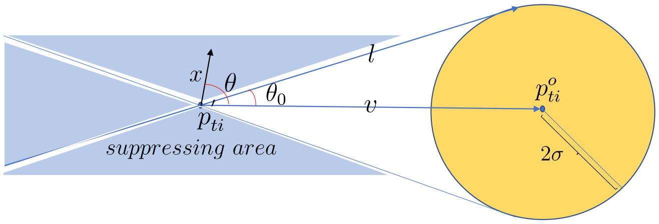


Fig. 2. Illustration of θ and θ_0, which are used in the definition of the flow norm. $\boldsymbol{x}$ is the derivative of the residual with respect to $\boldsymbol{p}'_{\boldsymbol{ti}}$, which totally depends on the local information. Conversely, θ_0 relies on global information $\boldsymbol{p}'_{\boldsymbol{ti}}$, $\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$ and σ.

IV. Approach

To deal with the local minima problem of direct alignment, we design a flow norm to guide the non-linear solver to jump out from the local minimum. Assume we have a coarse optical flow map between the image pair. The key idea of the FlowNorm is to balance the local information (residual decreasing direction) and global optical flow information. Because the optical flow is coarse and unreliable, we just down-weight those correspondences whose residual decreasing directions are highly inconsistent with the corresponding flow positions.

A. Flow Norm

Following the definition in Sect. III, $\boldsymbol{F}$ denotes the computed coarse flow map between I_s and I_t, $\boldsymbol{P}^{\boldsymbol{o}}_{\boldsymbol{t}}$ represents the flow positions computed by $\boldsymbol{P}^{\boldsymbol{o}}_{\boldsymbol{t}} = \boldsymbol{P_s} + \boldsymbol{F}$, and $\boldsymbol{P}'_{\boldsymbol{t}}$ is the projection position of $\boldsymbol{P_s}$ at I_t based on the current relative pose $\boldsymbol{T}$. $\boldsymbol{p_{si}}$, $\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$ and $\boldsymbol{p}'_{\boldsymbol{ti}}$ denote the ith item of $\boldsymbol{P_s}$,$\boldsymbol{P}^{\boldsymbol{o}}_{\boldsymbol{t}}$ and $\boldsymbol{P}'_{\boldsymbol{t}}$ respectively. The flow norm of residual e_i is defined as

$$L(\boldsymbol{p_{si}}, \boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}, \boldsymbol{p}'_{\boldsymbol{ti}}, e_i) = \begin{cases} e_i, & \left|\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}} - \boldsymbol{p}'_{\boldsymbol{ti}}\right|_2 \leq 2\sigma \\ e_i, & \cos\theta \leq \cos\theta_0 \\ (\frac{\cos\theta+1}{\cos\theta_0+1})e_i, & \cos\theta_0 < \cos\theta, \end{cases} \tag{6}$$

where σ is the variance of the computed flow (the method for computing σ is described in Sect. IV-B) and $|\cdot|_2$ denotes the L2 norm of the vector. $\boldsymbol{v} = \boldsymbol{p}'_{\boldsymbol{ti}} - \boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$ is the direction from projection position $\boldsymbol{p}'_{\boldsymbol{ti}}$ to the flow position $\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$, $\boldsymbol{x} = \frac{\partial e_i}{\partial \boldsymbol{p}'_{\boldsymbol{ti}}}$ represents the derivative direction at $\boldsymbol{p}'_{\boldsymbol{ti}}$, θ denotes the angle between $\boldsymbol{v}$ and $\boldsymbol{x}$, and $\cos(\theta)$ can be computed by

$$\cos(\theta) = \frac{\boldsymbol{v^T x}}{|\boldsymbol{v}|_2 \, |\boldsymbol{x}|_2}. \tag{7}$$

As shown in Fig. 2, when the projection position $\boldsymbol{p}'_{\boldsymbol{ti}}$ lies outside the circle with $\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$ as its center and 2σ as its radius, θ_0 represents the angle between the tangent line $\boldsymbol{l}$ and the direction $\boldsymbol{v}$. Thus

$$\cos(\theta_0) = \frac{\sqrt{\boldsymbol{v^T v} - \sigma^2}}{|\boldsymbol{v}|_2}. \tag{8}$$

In summary, we tend to activate these correspondences when their projection positions are close to the flow positions or local gradient agrees with global information.

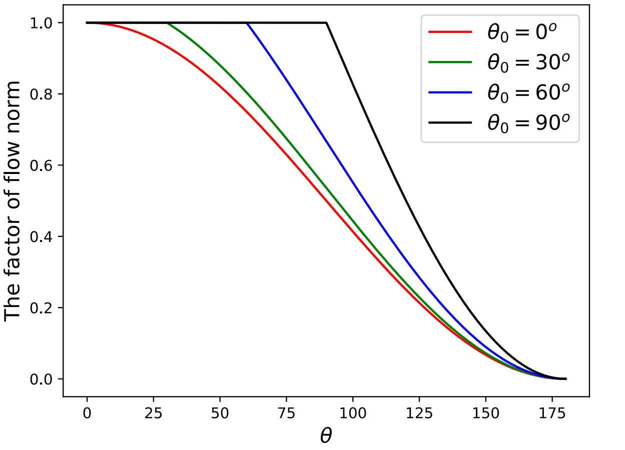


Fig. 3. Illustration of the change of the flow factor s with the increasing of θ.

With the proposed flow norm, the cost function of direct alignment is formulated as

$$\hat{\boldsymbol{\mathcal{X}}} = \arg\min_{\boldsymbol{\mathcal{X}}} \sum_{i}^{N} L(\boldsymbol{p_{si}}, \boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}, \boldsymbol{p}'_{\boldsymbol{ti}}, e_i). \tag{9}$$

The new optimal update step of the GN method for the jth iteration is

$$\Delta\widetilde{\boldsymbol{\mathcal{X}}_{\boldsymbol{j}}} = -(\boldsymbol{J}^{\boldsymbol{T}}_{\boldsymbol{j}} \boldsymbol{S} \boldsymbol{J_j})^{-1} \boldsymbol{J}^{\boldsymbol{T}}_{\boldsymbol{j}} \boldsymbol{S} \boldsymbol{E_j}, \tag{10}$$

where $\boldsymbol{S}$ is a diagonal matrix, and the ith row and ith column entry is the flow norm factor of the ith residual, which is summarized as

$$s_i = \begin{cases} 1, & \left|\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}} - \boldsymbol{p}'_{\boldsymbol{ti}}\right|_2 \leq 2\sigma \\ 1, & \cos\theta \leq \cos\theta_0 \\ (\frac{\cos\theta+1}{\cos\theta_0+1}), & \cos\theta_0 < \cos\theta. \end{cases} \tag{11}$$

Fig. 3 illustrates the change of the flow factor s with the increasing of θ. The θ_0 of these four lines is 0°, 30°, 60° and 90° respectively. From Eq. (8), for the same $\boldsymbol{p}'_{\boldsymbol{ti}}$ and $\boldsymbol{p}^{\boldsymbol{o}}_{\boldsymbol{ti}}$, a bigger θ_0 corresponds to a bigger flow uncertainty σ. For a big flow uncertainty, the flow norm will take more account of local information and assign larger weights to it. In case of the overshoot of the convergence process and the optimized results from being biased by the noise flow, we only involve the flow norm in a tracker when it runs at coarse levels of image pyramid. For example, the image pyramid of DSO has four levels, and we only involve our flow norm in the top two levels.

Although the form of the flow norm is similar to that of the Huber norm, their cores are very different. The Huber norm utilizes the local information of correspondences and the flow norm depends on the coarse flow. In fact, they are complementary to each other.

B. Shrunken PWC-Net

To obtain the optical flow map, we employ a shrunken PWC-Net to predict the optical flow between two images. Approaches that learn to predict optical flow from an image

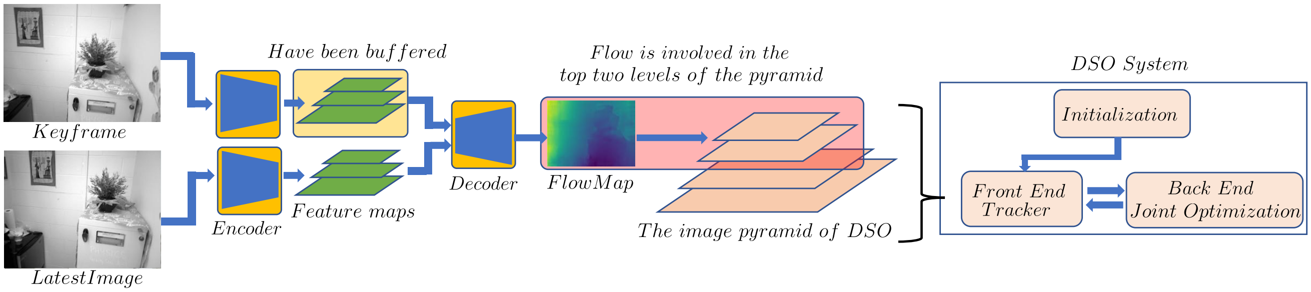


Fig. 4. Overview of the FlowNorm DSO. The predicted flow map is used to suppress those correspondences whose local gradients are highly inconsistent with predicted flow in the top two levels of the image pyramid.

pair have been studied in previous works [18]–[22]. However, due to the high computation cost, these previous nets cannot be migrated directly to our work. To efficiently obtain the optical flow, we choose the baseline network PWC-Net [21] as a reference network, and then shrink its convolutional layers and reduce its input image size. The shrinking process is a trade-off between prediction accuracy and computing efficiency. As our method works on the coarse levels of the image pyramid, it can robustly utilize the inaccurate optical flow.

Firstly, we change the input size of this network from $[3 \times 436 \times 1024]$ to $[1 \times 240 \times 320]$. RGB images should be transformed into grey images before feeding them into the shrunken network. Secondly, we remove one coding block and two pooling operations from the encoder, and the output size of the last encoding layer is $[15 \times 30]$. Finally, we remove one decoding block and reduce the correlation radius from 4 to 3, as the correlation operation of decoding block is computationally expensive. The size of the predicted flow is $[112 \times 160]$. Our encoding and decoding blocks are identical to the encoding and decoding blocks of PWC-Net. The shrunken network architecture is shown in the supplementary video.

Let Θ be the set of all the learn-able parameters in our shrunken network. W_Θ^l and W_{GT}^l denote the predicted flow field and the corresponding ground truth of the lth pyramid level respectively. We use the same multi-scale training loss proposed in FlowNet [19]:

$$\mathcal{L}(\Theta) = \sum_{l=l_0}^{L} \alpha_l \sum_{x} \left|W_\Theta^l(x) - W_{GT}^l(x)\right|_2 + \gamma|\Theta|_2, \qquad (12)$$

where the second term regularizes the parameters of the model in case of over fitting, the α_l and γ are the balance weights for different pyramid levels.

The variance σ of the predicted flow is computed by averaging the squared $L2$ error of the prediction results in the testing dataset.

C. Overview of the FlowNorm DSO

To demonstrate the effectiveness and efficiency of our method, we take our flow norm as a plug-in component for the baseline methods, DSO and BA-Net. The FlowNorm DSO is still a real-time system, which can be directly compared with DSO on any dataset.

As shown in Fig. 4, the plug-in is composed of three parts: the latest image is firstly fed into the encoder network to construct multi-scale feature maps. Then, the decoder will get the concatenation of the latest feature maps and the keyframe's feature maps and output a predicted flow map. Finally, the predicted flow map will be involved in the BA of the DSO tracker in the top two levels of the image pyramid. Although the pipeline needs to encode two images, all frames are only required to be encoded once by buffering the feature maps of the active keyframes.

D. Comparison with FlowInit DSO

To completely prove the effectiveness and efficiency of the flow norm, we also construct a competitive strategy. Given $\boldsymbol{p_{si}} \in \boldsymbol{P_s} = \{\boldsymbol{p_{si}}|i = 1 \cdots N\}$, $d_i \in D = \{d_i|i = 1 \cdots N\}$ at I_s and the predicted positions $\boldsymbol{p^o_{ti}} \in \boldsymbol{P^o_t} = \{\boldsymbol{p^o_{ti}}|i = 1 \cdots N\}$ at I_t, we compute the initial transform $\boldsymbol{T_0}$ by minimizing the geometric error

$$\boldsymbol{T_0} = \arg\min_{\boldsymbol{T}} \sum_{i}^{N} |\pi(\boldsymbol{p_i}, \boldsymbol{T}, d_i) - \boldsymbol{p^o_{ti}}|_2 \,, \qquad (13)$$

where $\pi(\cdot)$ is the projection function, and it is defined in Sect. III. Then, we take $\boldsymbol{T_0}$ as an initialization for the tracker of the DSO. We call the DSO with the initialization computed from the predicted flow map as FlowInit DSO. We find that the performance of this initialization strategy is on par with the FlowNorm DSO strategy for those well predicted flow positions. However, for very poor flow prediction, tracking with the initialization strategy is highly unstable. Because we shrink the size of the prediction network, our predicted flow usually has an overall offset. The overall offset causes that the initialization from the geometric BA also contains the offset. More comparison details are shown in the next Section.

V. Experiments

To verify the effectiveness of our method, we build FlowNorm versions of DSO and BA-Net. We evaluate our system on a Linux system with an Intel Core i7-7700 CPU of 3.50GHz and an Nvidia Titan Xp GPU.

A. *Training*

We train the shrunken PWC-Net on the SceneNN [13] dataset, which consists of 94 Kinect-captured RGB-D image sequences with ground truth poses. We select 44 / 25 image sequences from the SceneNN dataset and take them as training/testing sets respectively. Then, we sample pairs from the training and testing sets and generate the ground truth optical flow by projecting one pixel from one image to another image. During the projection process, we remove the occlusion area by verifying whether the depth of one pixel is consistent with the depth of its projection position.

Our shrunken PWC-Net is trained with ADAM [23] with the initial learning rate 0.0001. The weights in the training loss defined in Eq. (12) are set to be $\alpha 5 = 0.08$, $\alpha 4 = 0.02$, $\alpha 3 = 0.01$, and $\alpha 2 = 0.005$. The trade-off weight γ is set to be 0.0004. Although our network lacks some of the layers of PWC-Net, we still load the parameters of PWC-Net into the corresponding layers of the shrunken version as the initial parameters. The total training process takes one day on a computer with one Titan XP.

B. *FlowNorm in DSO*

We compare the FlowNorm DSO with the original DSO on two monocular datasets: TUM-MonoVO dataset [14] and ICL-NUIM dataset [15]. The TUM-MonoVO dataset provides 50 photometrically calibrated sequences, comprising different indoors and outdoors environments. The ICL-NUIM dataset contains 8 ray-traced sequences from two indoor environments. Since the TUM-MonoVO dataset only provides loop-closure ground-truth, we evaluate all sequences using the alignment error, which is defined in the TUM-MonoVO dataset.

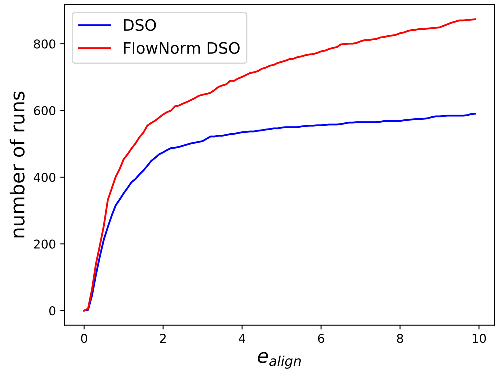
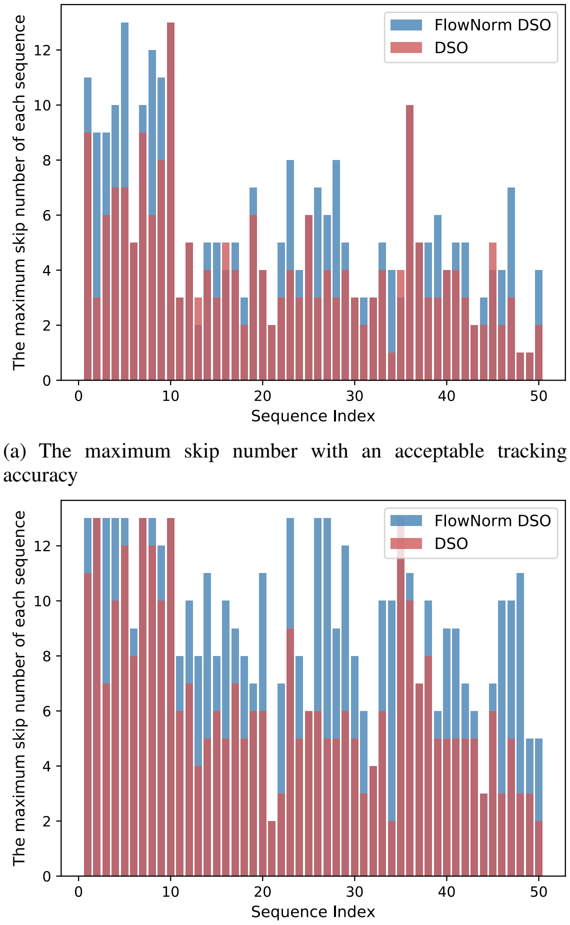


Fig. 5. The accumulated number of runs whose alignment errors are smaller than e_{align}; larger is better. Testing dataset contains all downsampled sequences from TUM-MonoVo and ICL-NUIM datasets.

To increase the difficulty of evaluation, we add a **new evaluation metric**. We downsample the image sequences with a skip of 1,2,3,4...13 frames. We track 2 times for every sequence. Which means the total number of runs is 1508. Apart from the alignment errors, we also measure two numbers: the maximum skip number without a losing tracking and the maximum skip number that can be tracked with an acceptable accuracy for every sequence. We take the alignment error of sequences without downsampling as a reference for whether tracking has acceptable accuracy or not and label it as $error_0$. If the alignment error of one downsampling sequence is smaller than three times its corresponding $error_0$, we mark the tracking result of the run as an acceptable tracking accuracy.



(a) The maximum skip number with an acceptable tracking accuracy

(b) The maximum skip number without a losing tracking

Fig. 6. Comparison of the convergence ability in all 50 sequences of the TUM-MonoVO dataset.

Fig. 5 illustrates the statistical performance of FlowNorm DSO and DSO. The accuracy of FlowNorm DSO is better than that of the original DSO. The performance of the original DSO is on par with its FlowNorm version when the downsampling rate is low. However, FlowNorm DSO presents more robust performance with the increase of the downsampling rate. Fig. 6 shows the maximum skip number with acceptable tracking accuracy and the maximum skip number without losing tracking for all sequences in the TUM Mono dataset. FlowNorm DSO (blue) has consistently better performance than the original version. Note that we only downsample sequences with 1 to 13 steps, the maximum step with 13 means we do not find losing tracking or the tracking results of all runs are acceptable in this sequence. Fig. 7 shows the tracking trajectories of FlowNorm DSO (green) and the original DSO (red) on the first sequence of

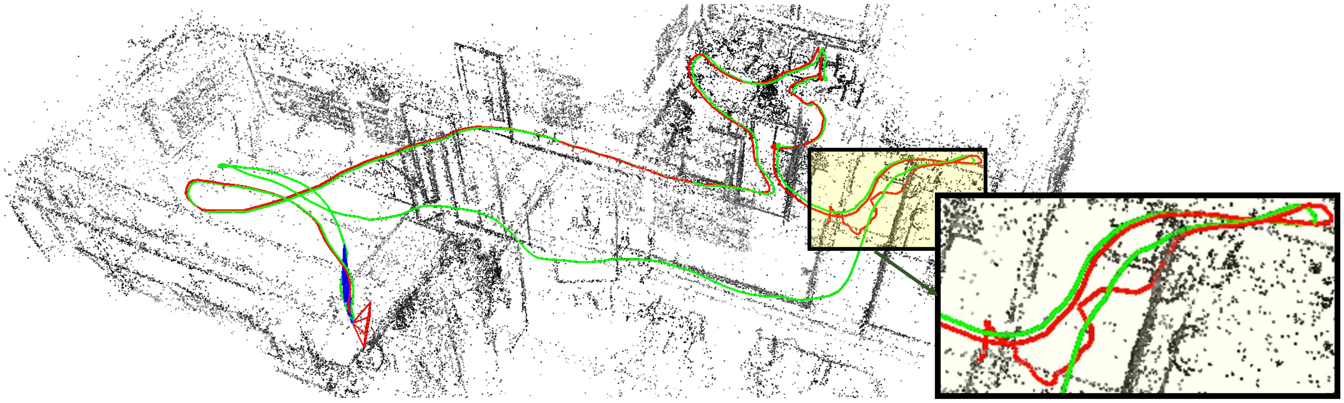


Fig. 7. An example of a losing tracking in the first sequence of the TumMonoVo dataset with a skip of 10 frames. The red and green trajectories are computed from DSO and FlowNorm DSO respectively.

the TUM Mono dataset with a downsampling rate of 10. DSO loses tracking in the black box area as the camera has large view change there.

C. FlowNorm in BA-Net

As BA-Net does not public codes, we construct a motion tracking version of it. The constructed BA-Net is trained on our training/validating dataset. Similar to FlowNorm DSO, we also use the predicted flow guide for the convergence of BA-Net. We use the remaining part of the SceneNN dataset to build a challenging image pair dataset. Then, we generate initial poses by adding rotation noise and translation noise to the ground truth poses of these image pairs. The final number of image pairs is 60126. We compare how many pairs are successfully aligned by BA-Net and its FlowNorm version. The results are 48261 and 37218 for FlowNorm version and the original version respectively, which proves our method can further expand the convergence range of the tracker based on minimizing the feature metric residual.

D. FlowNorm Vs FlowInit

To prove the effectiveness of the flow norm, we construct a competitive strategy, which is described in Sect. IV-D. We compute an initial pose from the predicted flow directly. We find that if we just take the computed initial pose as the initialization of the DSO tracker, the tracker becomes very unstable (usually loses tracking when it gets an inaccurate optical flow). We consider the reason for such losing tracking is that the indirect method seriously depends on the correct matching. However, the depths and correspondences used to compute the initial pose both contain a lot of noise. Next, we insert the computed initial pose to the queue of trying poses in the DSO tracker, and the queue in DSO is used to prevent loss of tracking. We compare the FlowInit DSO and FlowNorm DSO, and the results are shown in Table I. In the table, "accept. acc." and "w/o losing." mean the maximum skip number with an acceptable tracking accuracy and the maximum skip number without a losing tracking respectively. "Ave. align err." denotes the average of the first five alignment errors (downsampling rate from 1 to 5). Due to space limitation, we just show the comparison results for the first three sequences of the TUM-MonoVO dataset.

TABLE I

FlowNorm Vs FlowInit

Config	Seq.	Ave. align err.	accept. acc.	w/o losing.
DSO	01	0.5760	9	11
FlowInit	01	0.5380	9	13
FlowNorm	01	**0.5299**	**11**	13
DSO	02	1.0094	3	13
FlowInit	02	**0.3058**	8	13
FlowNorm	02	0.3765	**9**	13
DSO	03	0.7237	6	7
FlowInit	03	1.1205	7	8
FlowNorm	03	**0.6578**	**9**	13

E. Runtime analysis

In the implementation, we implement the trained model in DSO by PyTorch-C++[1], and we create a new thread for the flow prediction. The forward of the network is in the GPU and the other models of DSO are implemented in the CPU, which means the prediction of the flow does not have an effect on other models in DSO. In our computer, the forward process of the shrunken network takes 14 ms per frame.

VI. Conclusion and future work

In this paper, we have presented a flow norm to enhance the convergence range of direct alignment by utilizing a coarse flow map to constrain those correspondences that are highly inconsistent with the flow map. We employed a shrunken PWC-Net to generate the coarse flow map and built variants of DSO and BA-Net to prove the effectiveness of the flow norm. Meanwhile, we also compared the flow norm with a competitive strategy that gets the initial pose from the predicted flow directly. Our experiments proved the effectiveness and efficiency of the flow norm. In future work, we plan to investigate new network architectures to increase the accuracy of the prediction network and explore more formation of the flow norm.

VII. Acknowledgement

This work was supported by HKUST-DJI Joint Innovation Laboratory and Hong Kong PhD Fellowship Scheme.

[1] https://pytorch.org/cppdocs/

References

[1] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based SLAM. In *Asian Conference on Computer Vision*, pages 324–341. Springer, 2016.

[2] Amal Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3D modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 1486–1493, 2014.

[3] Jakob Engel, Thomas Schps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision(ECCV)*, pages 834–849. Springer, 2014.

[4] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, pages 1449–1456, 2013.

[5] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. In *2009 IEEE 12th International Conference on Computer Vision(ICCV)*, pages 72–79. IEEE, 2009.

[6] David Nistr. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756, 2004.

[7] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 4104–4113, 2016.

[8] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3057–3064. IEEE, 2011.

[9] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

[10] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. *In International Conference on Learning Representations(ICLR)*, 2019.

[11] Lukas von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The gauss-newton loss for multi-weather relocalization. 2019.

[12] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J. Davison. Learning to solve nonlinear least squares for monocular stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[13] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101. IEEE, 2016.

[14] Jakob Engel, Vladyslav Usenko, and Daniel Cremers. A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*, 2016.

[15] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531. IEEE, 2014.

[16] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2017.

[17] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[18] Jia Xu, Ren Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1297, 2017.

[19] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[21] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[22] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.