Incorporating learnt local and global embeddings into monocular visual SLAM

Huaiyang Huang[1] · Haoyang Ye[2] · Yuxiang Sun[3] · Lujia Wang[1] · Ming Liu[1]

Abstract

Traditional approaches for Visual Simultaneous Localization and Mapping (VSLAM) rely on low-level vision information for state estimation, such as handcrafted local features or the image gradient. While significant progress has been made through this track, under more challenging configuration for monocular VSLAM, e.g., varying illumination, the performance of state-of-the-art systems generally degrades. As a consequence, robustness and accuracy for monocular VSLAM are still widely concerned. This paper presents a monocular VSLAM system that fully exploits learnt features for better state estimation. The proposed system leverages both learnt local features and global embeddings at different modules of the system: direct camera pose estimation, inter-frame feature association, and loop closure detection. With a probabilistic explanation of keypoint prediction, we formulate the camera pose tracking in a direct manner and parameterize local features with uncertainty taken into account. To alleviate the quantization effect, we adapt the mapping module to generate 3D landmarks better to guarantee the system's robustness. Detecting temporal loop closure via deep global embeddings further improves the robustness and accuracy of the proposed system. The proposed system is extensively evaluated on public datasets (Tsukuba, EuRoC, and KITTI), and compared against the state-of-the-art methods. The competitive performance of camera pose estimation confirms the effectiveness of our method.

Keywords Visual simultaneous localization and mapping (SLAM) · Visual-based navigation · Mapping

1 Introduction

Visual Simultaneous Localization and Mapping (VSLAM) is a fundamental building block for robotic state estimation (Cadena et al. 2016), providing critical information for high-level applications. With the recent progress in this field, VSLAM has been well developed for various real-world applications, e.g., visual-based navigation for autonomous vehicles, ranging from path planning to decision making. Moreover, VSLAM can model the environment and build a corresponding map apart from ego-motion estimation. This capability can further benefit the robots in scene understanding and object recognition.

✉ Ming Liu
  eelium@ust.hk

  Huaiyang Huang
  hhuangat@connect.ust.hk

  Haoyang Ye
  hy.ye@connect.ust.hk

  Yuxiang Sun
  yx.sun@polyu.edu.hk; sun.yuxiang@outlook.com

  Lujia Wang
  eewanglj@ust.hk

[1] Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

[2] IAS BU Smart Driving Product Dept, Huawei Technologies, Shanghai, China

[3] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR, China

Despite the impressive progress under moderate scenarios, it is noticeable that diverse challenges (Sun et al. 2018), for instance, adverse illumination conditions (Park et al. 2017), could cause failure to most of the current VLSAM systems. Therefore, the robustness and accuracy under challenging situations remain a problem demanding further investigation. Emerging advances in deep learning (DL) provide an alternative perspective in resolving the aforementioned challenges. For examples, leveraging learning-based depth prediction (Mahjourian et al. 2018), object recognition (Liu et al. 2016), or high-level semantics (Long et al. 2015) benefits the visual state estimation in several aspects, varying from local reconstruction (Loo et al. 2019), scale recovery (Tateno et al. 2017; Yang and Scherer 2019) to dynamic environment handling (Runz et al. 2018). Especially in recent years, the exploration of learning local feature extraction and description is prevailing (Balntas et al. 2017). Following a *detect-and-describe* scheme to predict the repeatability and description within one forward pass, several works (DeTone et al. 2018; Dusmanu et al. 2019; Tang et al. 2019) demonstrate a superior efficiency against the traditional *detect-then-describe* pipelines, e.g., SIFT (Lowe 2004). The larger receptive fields from convolutional neural networks (CNNs) and metric learning techniques further contribute to the competitive performance of learning-based methods in both detection and description. These evolutions pave the way for the introduction of deep features into VSLAM systems. While some existing works generally recover pose in an end-to-end manner (Wang et al. 2017; Li et al. 2018), limiting the generalization ability for different scenarios, we consider a more practical approach to utilize the deep feature.

Built on top of ORB-SLAM2 (Mur-Artal and Tardós 2017), we propose a monocular SLAM system powered by learnt local and global embeddings. Leveraging the repeatability probability predicted from a CNN, the initial camera pose can be tracked robustly in a direct manner. Pre-triangulated landmarks can then be associated with local observations efficiently, which establishes correspondences for the pose refinement. We discuss the parameterization of local features to facilitate different modules from pose estimation to mapping. To achieve global consistency, we further leverage deep global embeddings to detect temporal loop closure.

Experimental results on public datasets, especially accurate pose estimations on challenging scenarios where state-of-the-art methods fail, demonstrate the effectiveness of the proposed system. Figure 1 provides an overview of our system. Our contributions are summarized as follows:

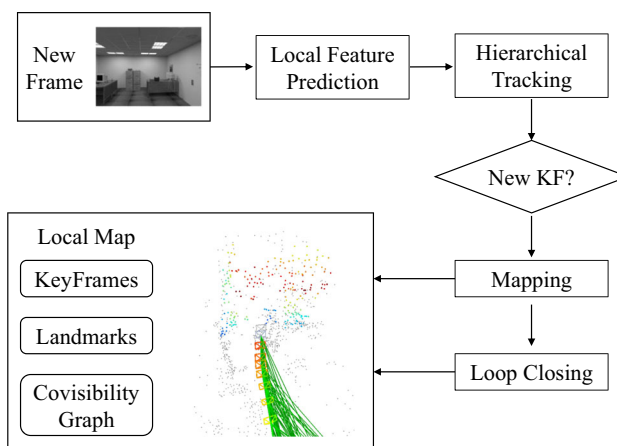– A monocular visual SLAM system leveraging learned local and global embeddings.

Fig. 1 System overview. Our system leverages deep networks to predict both local and global features. We follow the systematic design of ORB-SLAM2 (Mur-Artal and Tardós 2017), which can be divided into three modules: tracking, mapping and loop-closure. In the frontend, we propose a hierarchical scheme to recover the camera poses with network predictions. The mapping module generates new landmarks and maintains a covisibility graph. We further leverage learnt global embeddings to predict loop-closure and correct the global inconsistency

– A hybrid tracking scheme along with uncertainty modeling based on the network predictions.
– A mapping module adapted from the traditional pipeline to fit the nature of the frontend.
– Extensive evaluations on public datasets to demonstrate the accuracy and robustness of the proposed method.

Some parts of this work has been presented in Huaiyang Huang et al. (2020), In this paper, we extend the conference paper (Huaiyang Huang et al. 2020) with the following contributions:

– We improve the camera tracking scheme and provide a probabilistic explanation.
– We extend the previous visual odometry system into a complete SLAM system with global features predicted by a neural network.
– We perform more experiments to verify the effectiveness of different modules and the overall system.

The rest of the article is organized as follows: In Sect. 2, we review recent progress and challenges in VSLAM along with methods integrating deep learning. In Sect. 3, we demonstrate commonly used notations for simplicity. Then, Sects. 4, 5 and 6 describe the three main modules in our system, namely feature extraction, camera tracking and association and mapping and loop closing, respectively. In Sect. 7, we report the experimental results for the ablation and comparative studies. Finally, Sect. 8 concludes this paper.

2 Related works

2.1 Monocular visual SLAM and its challenges

The state-of-the-art approaches for monocular VO/VSLAM can be categorized into two dominant classes, namely *indirect* and *direct* methods. Indirect, or feature-based methods (Klein and Murray 2007; Strasdat et al. 2011; Mur-Artal et al. 2015), generally extract points of interest along with their descriptors as a sparse representation of the input image. With multiview correspondence association, camera motion and sparse structure are recovered via minimizing the reprojection error. Among these works, ORB-SLAM (Mur-Artal et al. 2015) exploits the covisibility relationships to strengthen the map reuse and frame management, balancing the accuracy and the computational demand.

Direct methods (Newcombe et al. 2011; Engel et al. 2014, 2018), on the contrary, directly minimize the photometric error on input images to track camera poses. In particular, DSO (Engel et al. 2018) optimizes camera poses, sparse scene structure, and camera model parameters jointly. To combine the advantages of both methods, Froster et al. propose SVO (Forster et al. 2014), which tracks the camera poses via sparse image alignment and utilizes hierarchical bundle adjustment (BA) as the backend for optimizing the structure and camera motion. Inspired by these works, we propose a hybrid scheme for the camera motion estimation, which tracks the pose initially on the predicted repeatability map and then refines it in an indirect manner. In the context of challenging conditions, robust VO/VSLAM remains unsolved (Cadena et al. 2016; Park et al. 2017; Yang et al. 2018). Potential solutions to these issues can be found in Alismail et al. (2016), Zhang et al. (2017) and Kim et al. (2017). These works typically specialize in a particular problem (e.g., handling high dynamic range (HDR) environment), which would introduce certain overhead under common scenarios (Alismail et al. 2016). In contrast, our system is developed for a general VSLAM system, which does not require any preprocessing on the camera input for the image quality enhancement. At the same time, in the experiments, it works well under most conditions.

2.2 Deep learning in visual SLAM

With the emergency of DL, recent researches demonstrate significant benefits by introducing relevant techniques into VSLAM systems. Compared with traditional methods in local feature detection (Harris et al. 1988; Shi and Tomasi 1993; Rosten and Drummond 2006) and description (Lowe 2004; Bay et al. 2006; Rublee et al. 2011), learning-based approaches (DeTone et al. 2018; Dusmanu et al. 2019; Revaud et al. 2019; Tang et al. 2019; Sarlin et al. 2019) exhibit a competitive performance in both matching accuracy and efficiency. Several works propose to use deep features in the camera pose estimation. Tang and Tan (2019) proposed BA-Net, where a feature-metric objective function is used for the direct image alignment, and the camera poses and predicted depth maps can be optimized in an end-to-end manner. In von Stumberg et al. (2020), Stumberg et al. proposed GN-Net, which introduced a Gauss-Newton loss for training the feature maps to be more invariant in the direct camera tracking. Works attempting to combine learnt local features with VO/VSLAM have been brought to our view (DeTone et al. 2018; Tang et al. 2019). In DeTone et al. (2018), with labels from a VO backend, the original SuperPoint (DeTone et al. 2018) is extended to predict the stability of local keypoints. In Tang et al. (2019), to enhance the frontend efficiency for onboard RGB-D VO, Tang et al. proposed GCNv2. They supervise the detector with ground-truth keypoint locations labeled by Shi-Tomasi score. To further train the descriptor with triplet loss, positive and negative matches are retrieved via the projective geometry. The above works generally focus on leveraging traditional methods (e.g., multiview geometry) to train a more practical and efficient frontend. On the contrary, here we consider bridging the gap of exploiting learned features to alleviate the challenges in monocular VSLAM.

3 Notations

Throughout the paper, we denote the image collected at the kth time as I_k and the corresponding frame as $\mathcal{F}_k$. The world frame $\mathcal{F}_w$ is set to be identical to the first camera frame $\mathcal{F}_0$.

For I_k, the rigid transform $\mathbf{T}_k \in \mathbf{SE}(3)$ maps a 3D landmark $\mathbf{x}_i \in \mathbb{R}^3$ in $\mathcal{F}_w$ to $\mathcal{F}_k$ using:

$$^{c_k}\mathbf{x}_i = \mathbf{R}_k\mathbf{x}_i + \mathbf{t}_k, \tag{1}$$

where $\mathbf{T}_k = [\mathbf{R}_k|\mathbf{t}_k]$. $\mathbf{R}_k$ and $\mathbf{t}_k$ are the rotational and translational components of $\mathbf{T}_k$, respectively. Accordingly, $^{c_k}\mathbf{x}_k$ denotes a 3D point in $\mathcal{F}_k$.

We denote a 2D pixel projected from a 3D landmark as $\tilde{\mathbf{u}}_{i,k}$. We use $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ to denote the projection function: $\tilde{\mathbf{u}}_{i,k} = \pi(^{c_k}\mathbf{x}_i)$, where $\tilde{\mathbf{u}}_{i,k}$ is the coordinate in the pixel coordinate. π is defined as $\pi(^{c_k}\mathbf{x}_i) = \mathbf{K}^{c_k}\mathbf{x}_i$, where $\mathbf{K}$ is the intrinsic matrix for pinhole model.

The update of a camera pose is parameterized as an incremental twist $\boldsymbol{\xi} \in \mathfrak{se}(3)$. We use a left-multiplicative formulation $\oplus : \mathfrak{se}(3) \times \mathbf{SE}(3) \rightarrow \mathbf{SE}(3)$ for the update of $\mathbf{T}_k$, which is denoted as:

$$\boldsymbol{\xi} \oplus \mathbf{T}_k := \exp(\boldsymbol{\xi}^\wedge) \cdot \mathbf{T}_k, \tag{2}$$

where the mapping $\mathbf{SE}(3) \rightarrow \mathbb{R}^6$ is defined by $\mathbf{T} = \exp(\boldsymbol{\xi}^\wedge)$, where $(\cdot)^\wedge$ is the wedge operator.

4 Learning-based feature extraction

Here we adopt SuperPoint (DeTone et al. 2018) as the feature extraction front-end. Recall the pipeline of SuperPoint, it first encodes the input image $I \in \mathbb{R}^{H \times W}$ with a single, shared encoder. Then two different heads decode a repeatability volume $\mathcal{H}' \in \mathbb{R}^{H_c \times W_c \times (C^2 + 1)}$ and a dense description $\mathcal{D}' \in \mathbb{R}^{H_c \times W_c \times 256}$, respectively. C is the size of the grid cell, which is 8 in our system, and $H_c = H/C$, $W_c = W/C$. $\mathcal{H}'$ is then normalized by channel-wise softmax:

$$\mathcal{H}(h_c, w_c, y) = \frac{\exp\left(\mathcal{H}'(h_c, w_c, y)\right)}{\sum_{k=1}^{65} \exp\left(\mathcal{H}'(h_c, w_c, k)\right)}. \tag{3}$$

In this architecture, the keypoint detection problem is then formulated as a classification problem. Specifically, for $\mathcal{H}(h_c, w_c, :) \in \mathbb{R}^{C^2+1}$, the first C^2 channels represent the probability of a pixel in a $C \times C$ grid to be a keypoint, while the last channel stands for the probability that no interest point lies in the current patch. For the details we refer the readers to DeTone et al. (2018).

We define the repeatability map, in patch-wise ($\mathcal{R}_d$) and in pixel-wise ($\mathcal{R}$) as:

$$\mathcal{R}_d = \mathcal{H}(., ., C^2 + 1), \ \ \mathcal{R} = -\log(s(\mathcal{H}(\cdot, \cdot, : -1))). \tag{4}$$

where '.' denotes the omitted channels, while ':' denotes "from the first channel to the last channel". $\mathbf{s}()$ denotes the function that shuffles a feature vector into a 2D patch. In other words, it flattens the 3D volume to a 2D map. Here we further define the patch location where the pixel $\mathbf{u}$ belongs to as $\mathbf{u}_p$. Then a probabilistic explanation is given by:

$$p(\mathbf{u}|I, \mathbf{w}) = \mathcal{R}(\mathbf{u}), \tag{5}$$

$$p(\mathbf{u}_p|I, \mathbf{w}) = 1 - \mathcal{R}_d(\mathbf{u}), \tag{6}$$

where $\mathcal{R}_d \in \mathbb{R}^{H_c \times W_c}$, the last channel of $\mathcal{H}$, stands for nonexistence of interest point in current patch. Accordingly, we reinterpret $\mathcal{R}_d$ as the patch-wise repeatability prediction, which is used for the initial camera tracking, described in Sect. 5.1. $\mathcal{R} \in \mathbb{R}^{H \times W}$ is the repeatability map with full-resolution and $s : \mathbb{R}^{H_c \times W_c \times C^2} \to \mathbb{R}^{H \times W}$ maps the volume to a 2D heatmap. In the above formulation, we negate the repeatability response, so that a pixel $\mathbf{u}$ (or patch $\mathbf{u_p}$) is more leaning to be a keypoint (or to contain a keypoint) if it has a smaller response $\mathcal{R}(\mathbf{u})$ (or $\mathcal{R}_d(\mathbf{u_p})$).

In a non-maximum suppression (NMS) scheme, the locations of 2D features along with the final 2d grid $\mathcal{O} \in \mathbb{R}^{H_c \times W_c}$ are extracted from $\mathcal{R}$. A single cell in $\mathcal{O}$ stores the index of the interest point in the current patch, zero if no salient point exists. For a local feature $\mathbf{u}_i$, the corresponding descriptor $\mathbf{d}_i$ is sampled from $\mathcal{D} \in \mathbb{R}^{H \times W}$, which is interpolated from

$\mathcal{D}'$. In addition, for the keyframes selected in our system, we predict the global embedding vector with NetVLAD (Arandjelovic et al. 2016), which is used to measure the image similarity across different frames for detecting the loop closure.

5 Camera tracking and association

5.1 Camera pose tracking

Assuming an initial sparse structure built from the backend, the frontend recovers the camera pose and associates 3D landmarks with 2D observations, which is illustrated in Fig. 2.

Initially, we track the current camera pose in a direct manner. The direct method of the camera pose estimation advances in their correspondence-free formulation, where there is no need for heuristic design of matching strategies. To take both the advantage of direct method and the network predictions, we propose to hierarchically track the camera pose on the repeatability map. Our intuition behind is that 3D landmarks triangulated from 2D observations are consistently repeatable in different views. In other words, the reprojection location of a 3D landmark in $\mathcal{R}_{(\cdot)}$ should be the local extremum.

For the simplicity, here we denote the set of visible landmarks as $\mathcal{X} = \{\mathbf{x}_0, \mathbf{x}_1 \ldots \mathbf{x}_n\}$. The current camera pose is denoted as $\mathbf{T}_k$ and is parameterized as $\boldsymbol{\xi}_k$. Given a landmark position $\mathbf{x}_i$ and the current camera motion parameter $\boldsymbol{\xi}_k$, the likelihood of local observation can be derived as:

$$p(\tilde{\mathbf{u}}_{ik}|\mathbf{w}, \boldsymbol{\xi}_k, \mathbf{x}_i) = \mathcal{R}(\tilde{\mathbf{u}}_{ik}) = \mathcal{R}(\pi(\exp(\boldsymbol{\xi}_k^\wedge) \cdot \mathbf{x}_i)). \tag{7}$$

To estimate $\boldsymbol{\xi}_k$, our method seeks to minimize the total likelihood, which is given by:

$$
\begin{aligned}
\hat{\boldsymbol{\xi}}_k &= \arg\max_{\boldsymbol{\xi}_k} \prod_{\mathbf{x}_i \in \mathcal{X}} p(\cdot) = \arg\min_{\boldsymbol{\xi}_k} - \sum_{\mathbf{x}_i \in \mathcal{X}} \log(p(\cdot)) \\
&= \arg\min_{\boldsymbol{\xi}_k} - \sum_{\mathbf{x}_i \in \mathcal{X}} \log\left(\mathcal{R}(\pi(\exp(\boldsymbol{\xi}_k^\wedge) \cdot \mathbf{x}_i))\right),
\end{aligned}
\tag{8}
$$

where we abbreviate $p(\tilde{\mathbf{u}}_{ik}|\mathbf{w}, \boldsymbol{\xi}_k, \mathbf{x}_i)$ to $p(\cdot)$. With a robust kernel function denoted as $\| \cdot \|_\gamma$, the total energy function is given by:

$$E = \sum_{\mathbf{x}_i \in \mathcal{X}} \| -\log(\mathcal{R}(\pi(\exp(\boldsymbol{\xi}_k^\wedge) \cdot \mathbf{x}_i)))\|_\gamma. \tag{9}$$

In the implementation, we choose Huber norm as the loss function. Based on the above derivation, this problem is converted into a standard least-squares problem, which can be solved using Gauss-Newton method.

Algorithm 1: Tracking and Feature Association

Data: $I_k, \tilde{\xi}_k, \mathcal{O}_k, \mathcal{R}, \mathcal{R}_d, \mathcal{M}$.

1 $\tilde{\xi}_k = \text{applyConstantVelocity}(\tilde{\xi}_k)$
2 $\mathcal{X} = \text{collectVisibleLandmarks}(\mathcal{M})$
3 $// \mathcal{M}$ denotes the local map.
4 $\hat{\xi}_k = \text{coarseTracking}(\tilde{\xi}_k, \mathcal{X})$
5 **foreach** $\tilde{u}_{i,k}$ **do**
6     $\mathcal{V}_{i,k} = \text{checkAdjacentPatches}(\tilde{u}_{i,k}, \mathcal{O}_k)$
7   **if** $|\mathcal{V}_{i,k}| \equiv 1$ **then**
8        $\text{associate}(x_i, \mathcal{V}_{i,k}[0])$
9   **else**
10        $j = \text{findBestAssociation}(x_i, \mathcal{V}_{i,k})$
11        $\text{associate}(x_i, \mathcal{V}_{i,k}[j])$
12   **end**
13 **end**
14 $\hat{\xi}_k = \text{poseRefinement}(\hat{\xi}_k, \mathcal{X})$
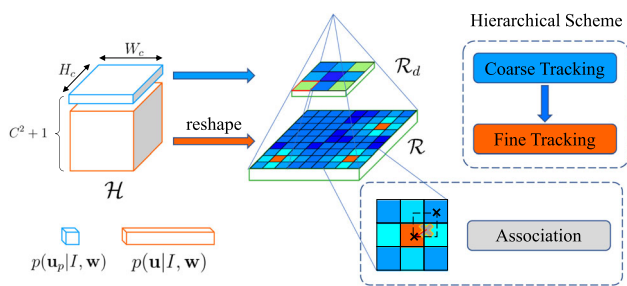15 $\text{outlierFiltering}(\hat{\xi}_k, \mathcal{X})$

Fig. 2 Tracking scheme of the proposed system. We reinterpret the local feature detection as a two-level feature probability heatmap. Our method then tracks the repeatability map in a coarse-to-fine manner. With the tracking result, we associate the features with landmarks locally

Direct method is considered very sensitive to the initial guess. A common strategy to alleviate this issue is to leverage a pyramidal implementation. In our case, generating pyramidal prediction results require multiple forward passes, prohibiting its real-time performance. However, we disassemble the prediction into two levels, which represents the repeatability from coarse to fine, as described in Sect. 4 and shown in the Fig. 2. Therefore, similar to the pyramidal implementation in the optical flow estimation or direct image alignment, we hierarchically track the current camera in a coarse-to-fine manner. For estimating the camera pose in the coarsest level, oppositely we minimize the probability that a patch does not contain keypoint, causing a little subtlety in the formulation. This gives:

$$
\begin{aligned}
\hat{\xi}_k &= \arg\min_{\xi_k} p(u_p | \xi_k, x_i, w), \\
&= \arg\min_{\xi_k} \mathcal{R}_d(\pi_d(\exp(\xi_k^\wedge) \cdot x_i)).
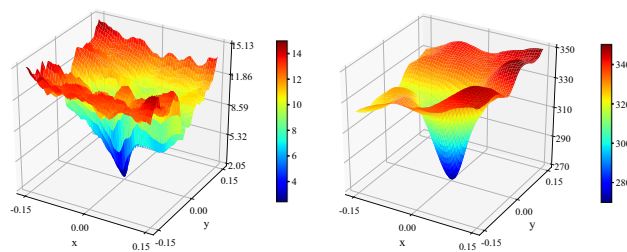\end{aligned}
\tag{10}
$$

Fig. 3 Cost surface for camera pose tracking of *photometry*-based residual (left) and *repeatability*-based residual (right). For each plot, the x–y plane stands for different translational offsets to the ground-truth pose and the value on z-axis is the total cost with the corresponding transform

For each residual term, the jacobian can be derived as:

$$
\begin{aligned}
J_i &= J_{\text{repeat}} \cdot J_{\text{proj}} \cdot J_{\text{pose}} \\
&= \frac{\partial \mathcal{R}_{(\cdot)}(\tilde{u}_{i,k})}{\partial \tilde{u}_{i,k}} \cdot \frac{\partial \tilde{u}_{i,k}}{\partial x_i^c} \cdot \frac{\partial \left((\xi_k \oplus T_k) x_i \right)}{\partial \xi_k},
\end{aligned}
\tag{11}
$$

where similar to direct tracking methods, J_{repeat} is the gradient of the corresponding repeatability map, which is evaluated at the projected pixel $\tilde{u}_{i,k}$ and is calculated by bilinear interpolation at each iteration. J_{proj} and J_{pose} are the Jacobian of projection function w.r.t transformed point and the left-compositional derivative of the the transformed point w.r.t the twist update ξ_k, respectively.

In practice, we find that tracking on the coarse level already produces a relatively good pose estimation result. When this estimation is fed to the fine level, our tracking module is able to fine-tune it for better accuracy. Analogous to the traditional optical flow pipeline, the pyramidal implementation is a significant design to alleviate the convergence basin issue and make the tracking module work better with predicted repeatability. Figure 3 compares the cost surfaces of our method and the classical direct method. The cost surface of photometry-based tracking suffers from non-convexity responsible for the sensitivity to initial guess, the narrowness of convergence basin, and potential numerical issues in the optimization. In contrast, tracking on repeatability exhibits a smoother cost surface, which in practice yields better convergence property.

5.2 Feature association

With the initial tracking, Here we propose a local feature association scheme, which minimizes the computational overhead by avoiding the descriptor matching as much as possible. Our observation is that, with the initial tracking, the reprojection location is considered to have an accuracy at sub-pixel level. That is to say, if the optimization converges, the reprojection location $\tilde{u}_{i,k}$ of an inlier x_i is supposed to have sub-patch accuracy. In addition, as we described before,

we perform NMS or griding to store one feature in a 8×8 patch. Under these observations, the 2D association of $\mathbf{x}_i$ should belonging to the four adjacent patches or grid cells of $\tilde{\mathbf{u}}_{ik}$. Therefore, instead of exhaustively matching descriptors or adopting some heuristics for tracking, we actively attempt to associate features in four adjacent patches. Based on the actual number of adjacent patches which contain a keypoint, we associate landmarks and local features as follows:

– If there is only one keypoint candidate, we directly associate it with the landmark.
– If there are multiple keypoint candidates, we then associate it according to descriptor distance, which is equivalent to finding the best local association.

It is noticeable that this association step is achieved with minimal computational overhead, which might be caused by matching high-dimensional descriptors.

5.3 Feature parameterization and pose refinement

For the indirect methods, the *reprojection error* is generally used for pose estimation or Bundle Adjustment. A generic formulation is given by:

$$\mathbf{u}_{i,k} = \mathbf{h}_{i,k}(\mathbf{x}_i, \boldsymbol{\xi}_k) + \mathbf{n}_{i,k}, \tag{12}$$

where $\mathbf{h}_{i,k}$ is the measurement function between the ith landmark position $\mathbf{x}_i$ and kth keyframe pose $\boldsymbol{\xi}_k$; and $\mathbf{n}_{i,k}$ is zero-mean white Gaussian noise that perturbs the measurement, i.e., $\mathbf{n}_{i,k} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{i,k})$. In other words, this formulation models local observations as 2D Gaussian components, i.e., $\tilde{\mathbf{u}}_{i,k} \sim \mathcal{N}(\bar{\mathbf{u}}_{i,k}, \bar{\boldsymbol{\Sigma}}_{i,k})$. As long as the majority of methods follow this generic formulation, the problem is how to define the parameters for each local feature, i.e., $\bar{\mathbf{u}}_{i,k}$ and $\bar{\boldsymbol{\Sigma}}_{i,k}$. Previous works resolve this issue in two approaches. In the early stage, methods model local features as 2D Gaussian distributions with anisotropic covariance. Some interesting discussions can be found in Kanazawa and Kanatani (2003) and Brooks et al. (2001), where the authors generally conclude modelling such anisotropic covariance can improve the visual state estimation. On the contrary, modern SLAM systems (Mur-Artal et al. 2015; Mur-Artal and Tardós 2017; Forster et al. 2014) generally assign an isotropic covariance for each feature. This gives $\bar{\boldsymbol{\Sigma}}_{i,k} = \sigma^2 \mathbf{I}_{2\times2}$, with σ proportional to the pyramidal level of the corresponding feature. However, as described above, generating a pyramidal prediction is prohibitive, making this parameterization method not practical in our implementation.

Following previous works, we seek to weigh the contribution of different local features to the overall estimation. Compared to traditional feature extraction techniques, CNNs have a larger receptive field and are capable of generating more consistent local features. Therefore, we model their 2D distribution from keypoint prediction heatmap with a probabilistic explanation. Considering the prediction as a mixture of multivariate Gaussian distributions, to approximate the single variate parameters, we have:

$$\bar{\mathbf{u}} = \mathbb{E}[\mathbf{u}] = \sum_{\mathbf{u} \in \mathcal{P}} \mathbf{u} \cdot p(\mathbf{u}|\mathbf{w}) \tag{13}$$

$$\begin{aligned} \bar{\boldsymbol{\Sigma}} &= \mathbb{E}[(\mathbf{u} - \bar{\mathbf{u}})(\mathbf{u} - \bar{\mathbf{u}})] \\ &= \sum_{\mathbf{u} \in \mathcal{P}} (p_i^2 \boldsymbol{\Sigma}_{\mathbf{u}} + p_i \mathbf{u}_i \mathbf{u}_i^T) - \mathbf{u}\mathbf{u}^T \end{aligned} \tag{14}$$

For the simplicity of implementation, here we represent $\mathcal{P}$ as a 3×3 patch. This parameterization scheme is further verified by Sect. 7.2.

After associating landmarks with the local observations, the current camera pose $\mathbf{T}_k$ is refined by minimizing the reprojection error. For $\mathbf{x}_i$, the error function is defined as:

$$\mathbf{e}_{i,k}^{\text{repro}} = \pi(\mathbf{R}_k \mathbf{x}_i + \mathbf{t}_k) - \tilde{\mathbf{u}}_{i,k}. \tag{15}$$

Similar to the initial tracking, the refined pose is solved by iterative least-squares. The overall energy function to minimize is defined as:

$$E^{\text{repro}} = \sum_{i \in \mathcal{P}_k} \| \left(\mathbf{e}_{i,k}^{\text{repro}} \right)^T \boldsymbol{\Sigma}_{i,k}^{-1} \mathbf{e}_{i,k}^{\text{repro}} \|_\gamma, \tag{16}$$

where $\boldsymbol{\Sigma}_{i,k}$ is the covariance of 2D feature location $\tilde{\mathbf{u}}_{i,k}$ for weighting different features' contribution to the optimization. The total pipeline of our tracking and association module is also demonstrated in Algorithm 1.

6 Mapping and loop closing

6.1 Mapping

6.1.1 Landmark creation

When a new keyframe is constructed, the mapping module first creates new landmarks with previous observations. For correspondence search, the top-n keyframes sharing the most covisibility score with the current keyframe are chosen to be the candidate frames. ORB-SLAM2 (Mur-Artal and Tardós 2017) exploits bag-of-words (BoW) (Gálvez-López and Tardós October 2012) to accelerate feature matching between keyframes for the landmark generation. Inevitably, this strategy introduces quantization effect that fewer correspondences would be found compared to brute-force search. For original ORB-SLAM2, we observe that such effect is affordable as it extracts quantities of features (e.g., 1000 for a image with VGA resolution). However, in our case, two

factors amplify this quantization effect: first, we only extract the most representative features, limiting the number of total local features; second, float descriptors lie in a much larger space than binary descriptors, making quantization effect more significant as more clusters are required for a representative quantization. This degrades both the robustness and the accuracy in practice. Therefore, two methods are adopted for feature association between keyframes. The first one is the approximate nearest neighbor (ANN) search. At the creation of each keyframe, a database of untracked local features is established. In the association step, we query each feature from the databases of candidate keyframes. Moreover, after the association step, the database is reindexed to guarantee the search efficiency with future keyframes. The problem of ANN-based association is that for the repetitive pattern (e.g. checkboard), ambiguity exists and increases the number of outliers in the triangulation step.

To resolve this problem, we further search the epipolar line of the 2d grid $\mathcal{O}_j$ of the target keyframe $\mathcal{F}_j$ to associate the features. For an untracked local feature $\mathbf{u}_{i,k}$ in the current keyframe $\mathcal{F}_k$, the epipolar line ${}^i\mathbf{l}_{k,j} = [l_0, l_1, l_2]^T$ on the image plane of $\mathcal{F}_j$ is derived as:

$$({}^i\mathbf{l}_{k,j})^T = \mathbf{u}_{i,k}^T \mathbf{K}^{-T} \mathbf{t}_{k,j}^\wedge \mathbf{R}_{k,j} \mathbf{K}^{-1},$$
$$\text{with} \quad \mathbf{R}_{k,j} = \mathbf{R}_k \mathbf{R}_j^{-1} \quad \mathbf{t}_{k,j} = -\mathbf{R}_k \mathbf{R}_j^T \mathbf{t}_j + \mathbf{t}_k,$$

where $\mathbf{K}$ is the projection matrix. We search $\mathcal{O}_j$ along the entire epipolar line ${}^i\mathbf{l}_{k,j}$. If current grid has an unmatched feature, we further check the epipolar distance $d_{i,k,l,j}$ as the geometry constraint:

$$d_{i,k,l,j} = \frac{1}{\det(\mathbf{\Sigma}_{i,k})} \frac{\mathbf{u}_{l,j}^T \cdot {}^i\mathbf{l}_{k,j}}{\sqrt{l_0^2 + l_1^2}},$$

which is weighed by $1/\det(\mathbf{\Sigma}_{i,k})$ for the consideration of uncertainty. The inlier with the best descriptor distance is considered as a successful match for the candidate feature. Finally, the 3D positions are recovered via mid-point triangulation. For rejecting outliers in the triangulation, we follow Mur-Artal et al. (2015) to further verify the sign of depth and the reprojection error in both keyframes.

6.1.2 Backend optimization

Similar to ORB-SLAM2, batched BA is utilized for backend optimization and map management. The variables in the local map for optimization consisting of updates to keyframe poses $\mathcal{T} = [\boldsymbol{\xi}_{k_0}, \boldsymbol{\xi}_{k_1}, \ldots, \boldsymbol{\xi}_{k_n}]$ and positions of landmarks $\mathcal{M} = [\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m}]$. We denote the full state vector as $\mathcal{X} =$ $[\mathcal{T}, \mathcal{M}]$, which is solved by:

$$\mathcal{X} = \arg\min_{\mathcal{X}} \sum_{\mathbf{T}_k} \sum_{\mathbf{x}_i} \text{obs}(i, k) \|(\mathbf{e}_{i,k}^{\text{repro}})^T \mathbf{\Sigma}_{i,k}^{-1} \mathbf{e}_{i,k}^{\text{repro}}\|_\gamma, \quad (17)$$

where $\text{obs}(i, k) = 1$ if $\mathbf{x}_i$ is observed by $\mathcal{F}_k$, $\text{obs}(i, k) = 0$ otherwise. Note that, as in ORB-SLAM2 (Mur-Artal and Tardós 2017), the poses of keyframes that share no observations of visible landmarks with the current keyframe are fixed to maintain a consistent scale estimation. The map management operations are inherited from Mur-Artal and Tardós (2017). Briefly, it culls redundant keyframes, removes outliers in the local BA and fuses the landmarks.

6.2 Loop closing

We extend the built-in loop-closure module to work with learnt global embeddings. When a keyframe is created, we generate the global embedding from NetVLAD (Arandjelovic et al. 2016). The advantage of using traditional BoW is that we can leverage the inverted file formulation for efficient keyframe retrieval. Yet in our case, the network directly outputs a global embedding vector instead of preserving the visual words information. Therefore, in our implementation, we directly compare the $l2$-distance between the global embeddings between the current keyframe and the past keyframes that does not share any common observations with the current keyframe. Then loop-closure candidates are considered to have embeddings close to that of the current keyframe. With the loop closure candidates, we further adopt the built-in modules of ORB-SLAM2 (Mur-Artal and Tardós 2017) to perform candidate verification and loop closure correction. Briefly, if the loop closure is detected successfully, it first corrects the **Sim**(3) pose graph generated by the minimal spanning tree from the global covisibility graph. Then, global BA is performed with a similar pipeline described in Sect. 6.

7 Experimental results

In this section, we first perform ablation studies on different modules in our system, including the proposed direct tracking frontend and the feature parameterization. Then, we report the general state estimation accuracy on several public datasets and compare our system against other popular methods. Finally, we provide several qualitative results along with runtime analysis to further demonstrate the effectiveness of the proposed method.
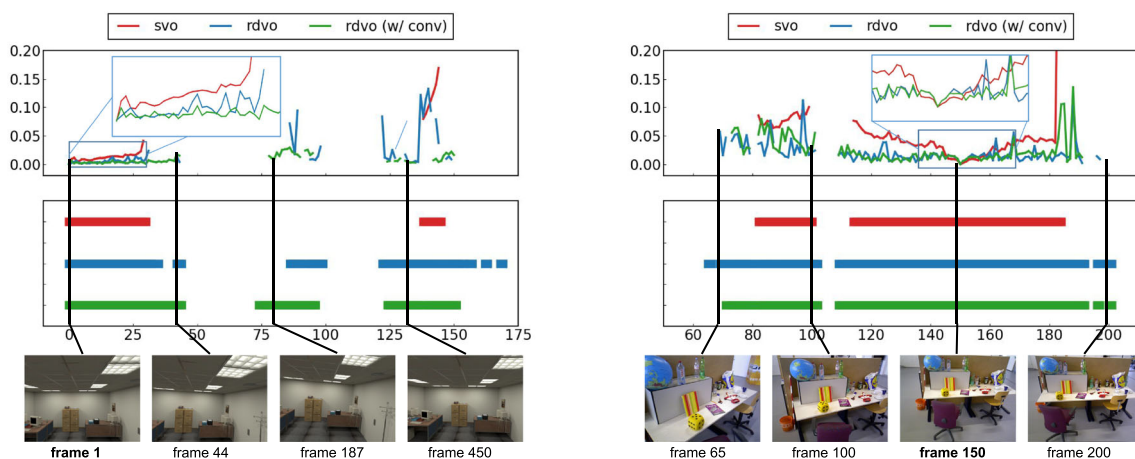
Fig. 4 Convergence radius and accuracy of different direct image alignment methods (indicated by color). All frames of the respective sequence are tracked on frame 1 (left) and frame 150 (right), using the identity as initialization. The bottom plots show for which frames tracking succeeds; the top plots show the final translational error. We observe that apart from the traditional photometry-based direct tracking (svo), the proposed repeatability-based tracking advances in larger convergence basin and better tracking precision (Color figure online)

7.1 Evaluation on the camera tracking

In this section, we validate the proposed direct tracking scheme. For the baselines to compare with, we first select the front-end of SVO (Forster et al. 2014). The tracking front-end of SVO detects FAST corners in a grid, and tracks recovered 3D landmarks in a sparse image registration manner, which shares a similar scheme to our system and is considered to be a good competitor. In addition, inspired by the recently proposed GN-Net in von Stumberg et al. (2020), we propose another baseline as competitor, which tracks the current frame with deep features as input in a direct manner. The objective function is given by:

$$E_{\text{feat}} = \sum \|\mathbf{F}_{k'}(\mathbf{u}_{i,k'}) - \mathbf{F}_k(\pi(\mathbf{R}_k\mathbf{x}_i + \mathbf{t}_k))\|, \qquad (18)$$

where $\mathbf{F}_k$ is the feature map of the kth frame. This formulation is based on the insight that CNNs can produce invariant feature for the correspondence across different frames. Similarly, we add this *feature-metric constraint* into our repeatability formulation. In the implementation, we leverage the feature volume predicted by SuperPoint and reduce its dimension using PCA to achieve an applicable real-time implementation on CPU. We denote this baseline as **rdvo (w/ conv)** and our method as **rdvo**.

Here we mainly evaluate the convergence radius and pose estimation accuracy of the tracking module. For the evaluation, given a test sequence, we first generate a sparse pointcloud with a pair of RGB image and depth map. Then, we evaluate different methods by registering the sparse visual structure to each frame in the sequence. The experiments are conducted on the ICL-NUIM (Handa et al. 2014) and TUM Dataset (Sturm et al. 2012). The results are shown in Fig. 4.

We observe that compared to SVO, the proposed method generally enlarges the convergence radius for direct tracking. In other words, starting from the same initialization point, more frames with larger translation and rotation variances are successfully registered by our method. This confirms the effectiveness of the proposed direct tracking scheme with predicted repeatability.

It is also interesting that, introducing high-level descriptions from the convolutional feature, rdvo (w/ conv) provides a close localization accuracy and robustness against rdvo. While both of them outperform the tracking scheme in SVO, rdvo (w/ conv) enlarges the convergence radius a little on the ICL-NUIM dataset, which indicates that using the *feature-metric* constraints can increase the robustness of direct tracking to some extend. Despite that the convergence radius is increased as expected, we notice that, rdvo (w/ conv) does not consistently outperform rdvo and svo. We think the reason could be that the image alignment generally requires a subpixel accuracy for an accurate alignment result. Unlike repeatability predictions that implicitly provide pixel-level keypoint location, convolutional feature slices of correspondences in different frames and have some subtleties with viewpoint variances. These subtleties yield some degradation to the registration results. Besides, although we interpolate between feature slices, the convolutional feature map is considered to provide patch-level information, which is downsampled 8 times in our case. As the residual dimension is very large (256 in our case), implementing it on CPU leads to significant computational overhead, especially for interpolating the feature map.

There exist some image alignment methods based on raw gradient input (Ling et al. 2018; Zhou et al. 2018; Nurutdinova and Fitzgibbon 2015; Kuse et al. 2016). However, these

methods differs from baselines in the experiments in that they densely select pixels with high gradient and generally transform the gradient map into another space, e.g., Distance Transform (DT), for smoother gradient information (Zhou et al. 2018; Kuse et al. 2016). Some also assume structural constraints between landmarks for better structural representation (Engel et al. 2014). As a consequence, we only focus on *sparse* image alignment methods here.

7.2 Evaluation on the local feature parameterization

In this section, we evaluate the effectiveness of different feature parameterization methods in pose estimation. As the local feature is parameterized as a gaussian distribution, with different combination of mean and variance, we mainly compare four different parameterization methods, given by:

- EXTREMA IDENTITY (EI): $\mathbf{u} = \mathbf{u}_{\text{peak}}, \boldsymbol{\Sigma} = \mathbf{I}_{2\times2}$
- EXTREMA COVARIANCE (EC): $\mathbf{u} = \mathbf{u}_{\text{peak}}, \boldsymbol{\Sigma} = \bar{\boldsymbol{\Sigma}}$
- MEAN IDENTITY (MI): $\mathbf{u} = \bar{\mathbf{u}}, \boldsymbol{\Sigma} = \mathbf{I}_{2\times2}$
- MEAN COVARIANCE (MC): $\mathbf{u} = \bar{\mathbf{u}}, \boldsymbol{\Sigma} = \bar{\boldsymbol{\Sigma}}$

where the $\bar{\mathbf{u}}$ and $\bar{\boldsymbol{\Sigma}}$ are described in Sect. 5.1.

For the dataset selection, we notice that the accuracy of SLAM system (or BA) is mainly bounded by the data association, and thus the local feature parameterization is not the most important factor. Therefore, using a real-world dataset can hardly compare the effectiveness of different parameterization methods. This is also because of the systematic error in the real-world dataset, for example, the uncertainty of calibration between cameras and sensors providing absolute pose measurements. As a consequence, here we perform the evaluations on the ICL-NUIM dataset.

To examine how different parameterization methods work in the SLAM system, for each frame in a sequence, we back-project the detected keypoints to the 3D space as landmarks. Then these landmarks are associated with N (5 in the experiment) adjacent frames. Combining the poses along with the landmark positions, we jointly optimize these variables following a standard pipeline of BA. We maintain an optimization problem with identical optimizable parameters and constraints, and introduce different local feature parameterization methods to examine their effectiveness.

The results are represented in Fig. 5, from which we have the following conclusions: First, the results show that considering the prediction with an anisotropic covariance assigning to different keypoints could generally increase the bundle adjustment accuracy. For methods that do not consider the uncertainty (EI and MI), the covariance matrices are all set to be identity. This means that in the optimization, different residual factors contribute to the final results equally. On the contrary, for methods taking uncertainty into account (EC and MC), the anisotropic covariance serve as a reflec-
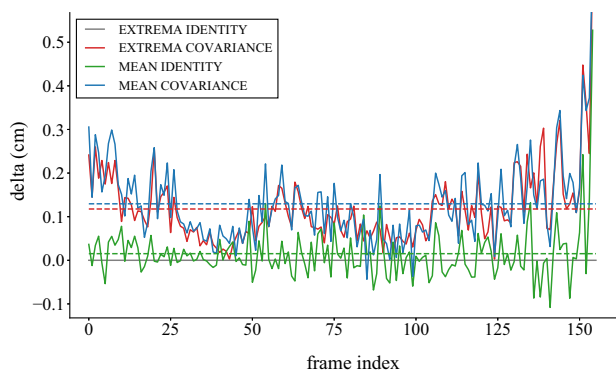

Fig. 5 Relative improvement on different methods compared to the baseline method. The dashed line represents mean error of the corresponding method

tion of keypoint prediction uncertainty. Through modelling the prediction uncertainty, different factors are weighed in the joint optimization according to their observation confidence. In other words, factors with higher uncertainty should have less contribution to the optimization framework. As a consequence, comparing EC and MC against EI and MI, the accuracy of BA is increased with modelling the uncertainty. Second, comparing methods that consider keypoint location as extrema (EI and EC) or weighted mean (MI and MC), we find that the pose accuracy is improved slightly. This indicates that most of the keypoint predictions are already of good quality, so that in a BA framework, considering the refinement of keypoint location with prediction uncertainty can only provide trivial improvement in the pose estimation. Accordingly, these experimental results guide us to finally choose the local feature parameterization in our system described in Sect. 5.1.

7.3 Evaluation on trajectory estimation

In this section, we evaluate our system on several public datasets, the New Tsukuba (Martull et al. 2012), the EuRoC Mav dataset (Burri et al. 2016) and KITTI Odometry dataset (Geiger et al. 2012). We compare our method to both the state-of-the-art indirect (ORB-SLAM2 Mur-Artal and Tardós 2017) and direct (DSO Engel et al. 2018) VO/VSLAM algorithms. GCN-SLAM (Tang et al. 2019), the recently proposed method using a learned binary descriptor, is expected to be one of the competitors. However, the monocular version adapted from the open-source implementation[1] fails to produce competitive results. Thus we exclude it for further evaluation. A major reason for the failure of GCN-SLAM is that it is designed for RGB-D inputs, while for the monocular VO, maintaining the scale consistency and estimating the depth of local feature raise more challenges.

---

[1] https://github.com/jiexiong2016/GCNv2_SLAM.

Table 1 Translational RMSE (cm) and tracking success rate (%) on the New Tsukuba dataset

Sequence name	Ours	DSO	ORB-SLAM w/o loop
Fluorescent	**10.9 ± 5.7**	59.0 ± 39.0	(18.2 ± 15.3)
	87.6 ± 0.3	98.2 ± 0.0	88.4 ± 11.6
Daylight	**9.4 ± 4.9**	50.0 ± 29.7	14.7 ± 9.7
	96.2 ± 0.2	98.1 ± 0.0	82.4 ± 16.9
Lamps	**9.3 ± 4.6**	✗	✗
	94.5 ± 7.1	44.5 ± 37.4	0.0 ± 0.0
Flashlight	**18.3 ± 10.3**	✗	✗
	94.4 ± 0.1	58.4 ± 10.9	0.0 ± 0.0

Best results are highlighted in bold

The tracking success is defined as (#tracked frame/#frame in the sequence)

All the experiments are done using the same desktop with i7-8700K and NVIDIA 1080Ti. For quantitative evaluation, translational RMSE of absolute trajectory error (ATE) (Sturm et al. 2012) is used. We run different algorithms on each sequence 5 times and average the evaluation metrics. The tracking failure is either reported by the system itself or determined afterward if the error is larger than 1 meter (typically caused by scale inconsistency). If tracking failure exists for one or more runs, the corresponding result is shown in parentheses (·), while failure for all runs is marked as ✗. The results on both datasets are reported in Tables 1, 2 and 3, respectively.

7.3.1 The new Tsukuba dataset

The New Tsukuba dataset provides synthetic images rendered by computer graphics techniques. It is challenging for monocular visual odometry as (1) the illumination condition of sequences *lamps* and *flashlight* is extreme, and (2) the camera rotation is relatively aggressive. As ORB-SLAM2 does not provide the setting for the Tsukuba dataset, we use the setting of another indoor dataset with the same image resolution and adjust the camera parameters accordingly. Note that the New Tsukuba Dataset does not contain any loopy motion, therefore all methods can be considered as pure visual odometry rather than full SLAM systems.

Table 1 reports the translational RMSE and success rate of tracking for each sequence. As expected, challenging illumination conditions of the *lamps* and *flashlight* lead both ORB-SLAM and DSO to tracking failure. Besides, to accelerate feature association for camera pose tracking, ORB-SLAM searches correspondences in a local window with a motion prior, making it sensitive to aggressive rotation change. As a consequence, it occasionally fails even under the *fluorescent* with a moderate illumination configuration.

Besides the robustness and accuracy under these test scenarios, our system is capable of maintaining a relatively

consistent trajectory estimation accuracy, regardless of the illumination variances. Especially on *lamps*, the RMSE is even slightly smaller than on *daylight*. Larger error under *flashlight* over other sequences indicates that learning-based descriptors share a similar characteristic with hand-crafted ones in degradation under photometric noise (Engel et al. 2018).

7.3.2 EuRoC Mav dataset

The EuRoC dataset contains several sequences that are challenging for monocular ego-motion estimation. Accordingly, for the comparison, we divide the dataset into *moderate* (easy) and *challenging* (hard) sequences. For the sequences MH01, MH02, MH03, V101, V201 and V202, the camera motion is slow, and the illumination conditions are moderate, making them relatively easy for monocular VO/VSLAM. On the contrary, in MH04, MH05, and V103, the illumination varies in a wide range. In V102, V103, and V203, the camera motion is aggressive, including nearly pure rotation or fast movement. Additionally, for that currently a pinhole camera model is assumed by our system, we pre-rectify all the images in the evaluation, which limits the field-of-view (FOV) of the camera and brings more challenges for monocular VO/VSLAM, especially indirect methods.

Here we report performances for both VO/VSLAM systems. Due to all the methods fail on V203, the corresponding results are not included. For the pure VO, as shown in Table 2, for the moderate sequences, the proposed system has comparable accuracy with the state-of-the-art methods. For the challenging sequences, our method increases the robustness and accuracy compared to the baselines. Especially for V103, ORB-SLAM is unable to track the camera motion with severe exposure change and aggressive motion continuously, while our method only fails in 1/10 runs, indicating a more robust performance. Considering the full SLAM system, the proposed system successfully increase the robustness and accuracy compared to the VO-only solution. It is noticeable that for the difficult sequence V103, we achieve an RMSE of 4.44 cm, which significantly improves the result from the pure VO system. Compared our method against ORB-SLAM2, we observe our system generally provides a comparable performance with improvement on V103.

7.3.3 KITTI odometry dataset

The KITTI Odometry Dataset (Geiger et al. 2012) is commonly used for evaluating visual state estimation systems in large-scale outdoor scenario. The result comparisons are reported in Table 3 for both pure odometry and SLAM. As in Table 3, the pure visual odometry results do not outperform ORB-SLAM2 while the full SLAM results is more competitive, which validate the improvement on introducing loop

Table 2 Translational RMSE (cm) on the EuRoC dataset

	Sequence	Ours w/o loop	DSO	ORB-SLAM2 w/o loop	Ours	ORB-SLAM2
Easy	Machine Hall 01	**1.63 ± 0.86**	5.73 ± 2.28	1.77 ± 0.91	–	–
	Machine hall 02	**1.46 ± 0.77**	4.53 ± 2.71	1.72 ± 0.81	–	–
	Machine hall 03	3.20 ± 1.53	20.89 ± 10.23	**3.19 ± 1.52**	–	–
	Vicon room 1 01	3.34 ± 4.34	13.52 ± 8.72	**3.28 ± 1.20**	–	–
	Vicon room 2 01	**1.85 ± 0.83**	4.70 ± 2.41	2.59 ± 1.34	**1.60 ± 0.15**	4.44 ± 1.31
	Vicon room 2 02	**4.73 ± 2.49**	12.31 ± 6.37	(5.08 ± 2.95)	2.59 ± 0.37	**1.69 ± 0.10**
Hard	Machine Hall 04	**8.59 ± 3.37**	20.03 ± 8.96	9.78 ± 4.05	–	–
	Machine hall 05	**4.34 ± 2.09**	10.47 ± 3.78	(12.67 ± 5.86)	–	–
	Vicon room 1 02	**23.82 ± 9.38**	32.83 ± 27.41	(14.06 ± 5.27)	**1.45 ± 0.11**	1.59 ± 0.35
	Vicon room 1 03	(17.17 ± 10.71)	**94.15 ± 39.78**	✗	**4.44 ± 1.31**	7.26 ± 5.78

Best results are highlighted in bold

Left column: performances of pure visual odometry. Right column: performance of SLAM system, where results on sequences without explicit loop-closure are excluded

Table 3 Translational RMSE (m) on the KITTI Odometry dataset

Sequence	Environment	Ours w/o loop	DSO	ORB-SLAM2 w/o loop	Ours	ORB-SLAM2
00	Urban	96.23 ± 44.76	118.58 ± 0.51	**73.39 ± 32.66**	**5.31 ± 0.57**	6.11 ± 0.32
01	Highway	✗	**105.14 ± 204.86**	✗	–	-
02	Urban+Country	76.94 ± 50.34	122.67 ± 6.93	**27.14 ± 14.04**	**22.54 ± 1.46**	28.92 ± 4.26
03	Country	1.34 ± 0.29	2.08 ± 0.08	**0.96 ± 0.53**	–	–
04	Country	**0.21 ± 0.08**	0.99 ± 0.01	1.14 ± 0.58	**0.51 ± 0.18**	0.87 ± 0.31
05	Urban	48.74 ± 26.33	50.93 ± 0.69	**39.42 ± 21.05**	**4.75 ± 0.61**	5.65 ± 1.04
06	Urban	**54.37 ± 25.51**	65.81 ± 1.29	54.45 ± 25.21	**10.16 ± 1.74**	16.13 ± 0.83
07	Urban	16.96 ± 9.37	18.37 ± 0.54	**16.71 ± 9.47**	4.16 ± 0.63	**2.45 ± 0.31**
08	Urban+Country	64.82 ± 39.95	119.12 ± 12.74	**51.55 ± 31.78**	–	–
09	Urban+Country	60.82 ± 31.60	61.21 ± 7.72	**47.31 ± 25.96**	**9.52 ± 0.47**	51.63 ± 12.70
10	Urban+Country	**8.27 ± 0.86**	34.84 ± 38.41	9.56 ± 1.02	–	–

Best results are highlighted in bold

closure with NetVLAD. Our system takes more time in data association due to the usage of floating descriptors, which limits the computation time for batched optimization. However, as more representative local features are selected, our system manages a more lightweight map and advances in global BA with loop closure. We notice that for the sequences with loops (e.g., 00, 02, 05-07, 09, 10), generally the proposed system performs better than ORB-SLAM2. Especially for sequence 09, we notice that ORB-SLAM2 failed to detect loop-closure with BoW, leading the trajectory to be inconsistent globally. On the contrary, our system successfully detects the loop with learnt global embedding. Along with global BA to optimize the trajectory, not surprisingly we achieve a better estimation result. Exception occurs on 08, where our system exhibits more severe scale drift in trajectory estimation, resulting in larger trajectory error. For the sequence 01, both methods fail to maintain a correct scale estimation, while only DSO succeeds.

7.4 Qualitative results

Here we also provide some qualitative results in Figs. 6, 7 and 8. They show the reconstructed sparse map and estimated trajectories on several sequences. In Fig. 6, even some repetitive features can be well associated and triangulated with relatively accurate depth, for example, features on the floor. As in Fig. 7, in this indoor environment, we notice that our estimated trajectory has only trivial drift without using loop-closure. In addition, the sparse map well recovers the geometry in the test desk scene. Figure 8 shows that the trajectory along with recovered visual structure on TUM Dataset, where we notice the estimated trajectory is well aligned with the ground truth.
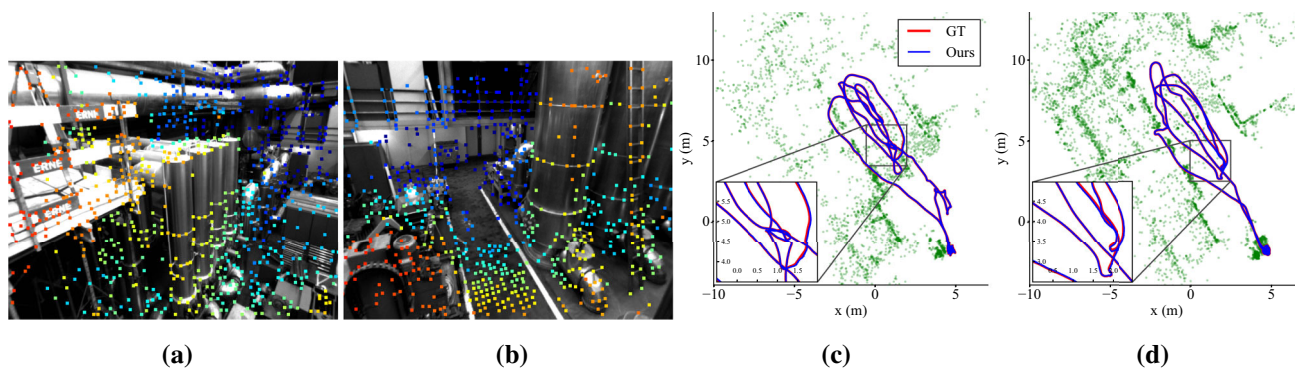
Fig. 6 **a**, **b** Excerpts of sequnce Machine Hall 01 and 02 the EuRoC dataset, keypoints are colored by the inverse depth. **c**, **d** trajectories of ours against the ground truth on Machine Hall 01 and 02 (Color figure online)
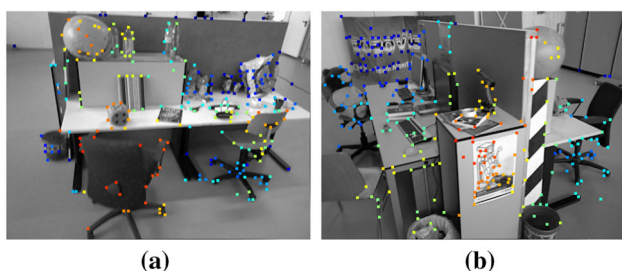

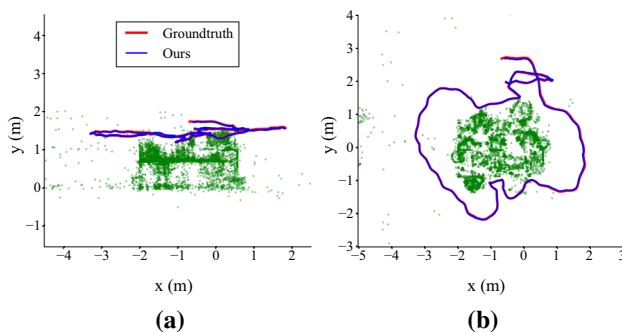**Fig. 7** Excerpts on the TUM Dataset, keypoints are colored by the inverse depth

Fig. 9 Evaluation on the local reconstructions. The higher the percentage of points towards a zero distance the better

Fig. 8 Qualitative results on TUM dataset

7.5 Evaluations on local reconstruction

To evaluate the reconstruction accuracy, the sparse point-clouds generated by different methods are scaled and transformed via the alignment results. The error metric is defined as the RMSE of distances to the nearest neighborhood. The similar evaluation process can be found in Millane et al. (2018). Three sequences V101, V201, and V202 of EuRoC that provide dense ground-truth structure are selected for the evaluation.

As shown in Fig. 9, our method recovers a more accurate local structure, which in turn guarantees the accuracy of local trajectory estimation. Additionally, in V101, although the ATE of our method does not outperform ORB-SLAM2,
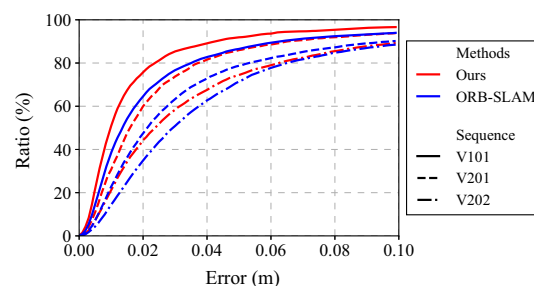
the local structure is more accurate than ORB-SLAM2, indicating fewer outliers in our system. One advantage of indirect methods over direct ones is the covisibility connections established by powerful descriptors, with which local map is fully reused and the VO drift can be reduced (Engel et al. 2018). ORB-SLAM2 does a great job of associating features cross views and fusing redundant points. However, under the cases of wider baseline, our system better associates landmarks with local features. Especially, comparing the sparse reconstruction of certain objects in the scene (e.g. the checkboard), drastically fewer outliers and redundancies are generated from our system.

8 Conclusions and future work

In this paper, we presented a monocular SLAM system leveraging learnt description, repeatability and global embedding. Different from previous work, we focused on tightly coupling the learning-based frontend with indirect multiview geometry to fully exploit the network predictions. By interpreting the repeatability prediction in a probabilistic manner, w proposed a two-step tracking scheme to estimate the camera pose: direct tracking on the repeatability maps and refining the pose with the patch-wise association. We further

discuss the local feature parameterization to consider the uncertainty in the network prediction. With loop closure detection from deep global embeddings, we maintain the consistency via global bundle adjustment. The experimental results demonstrated that the proposed system is capable to handle challenging situations, where both the state-of-the-art indirect and direct methods suffer from strong degradation.

In the future, we would like to investigate more lightweight network architectures or binary descriptor learning to accelerate our system. In the current system implementation, descriptor matching causes certain overhead for the map management, which we believe can be further optimized. In addition, supervising the learning frontend in an end-to-end manner might help our system better exploit from data-driven approaches. For example, using learnt relative pose as a prior for camera pose tracking could boost the general robustness under challenging conditions, in which cases direct formulations could not work well. Last but not least, extending the current system into a stereo or visual-inertial system is considered to be of much more practical value. As they can provide absolute scale information, such extension could better help robotic navigation in the long-term operations.

References

Alismail, H., Kaess, M., Browning, B., & Lucey, S. (2016). Direct visual odometry in low light using binary descriptors. *IEEE Robotics and Automation Letters*, *2*(2), 444–451.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307.

Balntas, V., Lenc, K., Vedaldi, A., & Mikolajczyk, K. (2017). Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5173–5182.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, Springer, pp. 404–417.

Brooks, M. J., Chojnacki, W., Gawley, D., & Van Den Hengel, A. (2001). What value covariance information in estimating vision parameters? In *Proceedings eighth IEEE international conference on computer vision, ICCV 2001*, Vol. 1, IEEE, pp. 302–308.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., & Siegwart, R. (2016). The euroc micro aerial vehicle datasets. In *The international journal of robotics research*. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Self-improving visual odometry. arXiv preprint arXiv:1812.03245.

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., & Sattler, T. (2019). D2-net: A trainable cnn for joint detection and description of local features. arXiv preprint arXiv:1905.03561.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, Springer, pp. 834–849.

Engel, J., Koltun, V., & Cremers, D. (2018). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 611–625.

Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, pp. 15–22.

Gálvez-López, D., & Tardós, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188–1197.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, IEEE, pp. 3354–3361.

Handa, A., Whelan, T., McDonald, J., & Davison, A. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE international (Ed.), ICRA, Hong Kong, China: conference on robotics and automation*.

Harris, C. G., Stephens, M. et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, Vol. 15, No. 50, Citeseer, pp. 10–5244.

Huaiyang Huang, Y. S., Ye, H., & Liu, M. (2020). Monocular visual odometry using learned repeatability and description. In *IEEE international conference on robotics and automation (ICRA)*, IEEE.

Kanazawa, Y., & Kanatani, K. (2003). Do we really have to consider covariance matrices for image feature points? In *Electronics and communications in Japan (part III: Fundamental electronic science)*, Vol. 86, No. 1, pp. 1–10.

Kim, P., Coltin, B., Alexandrov, O., & Kim, H. J. (2017). Robust visual localization in changing lighting conditions. In *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, pp. 5447–5452.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proceedings of the 2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE Computer Society, pp. 1–10.

Kuse, M., & Shen, S. (2016). Robust camera motion estimation using direct edge alignment and sub-gradient method. In *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, pp. 573–579.

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

Ling, Y., Kuse, M., & Shen, S. (2018). Edge alignment-based visual-inertial fusion for tracking of aggressive motions. *Autonomous Robots*, *42*(3), 513–528.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: single shot multibox detector. In *European conference on computer vision*, Springer, pp. 21–37.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Loo, S. Y., Amiri, A. J., Mashohor, S., Tang, S. H., & Zhang, H. (2019). CNN-SVO: improving the mapping in semi-direct visual odometry using single-image depth prediction. In *2019 International conference on robotics and automation (ICRA)*, IEEE, pp. 5218–5223.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.

Mahjourian, R., Wicke, M., & Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5667–5675.

Martull, S., Peris, M., & Fukui, K. (2012). Realistic cg stereo image dataset with ground truth disparity maps. In *ICPR workshop TrakMark2012*, Vol. 111, No. 430, pp. 117–118.

Millane, A., Taylor, Z., Oleynikova, H., Nieto, J., Siegwart, R., & Cadena, C. (2018). C-blox: A scalable and consistent tsdf-based dense mapping approach. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, pp. 995–1002.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5), 1147–1163.

Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). Dtam: Dense tracking and mapping in real-time. In *2011 International conference on computer vision*, IEEE, pp. 2320–2327.

Nurutdinova, I., & Fitzgibbon, A. (2015). Towards pointless structure from motion: 3d reconstruction and camera parameters from general 3d curves. In *Proceedings of the IEEE international conference on computer vision*, pp. 2363–2371.

Park, S., Schöps, T., & Pollefeys, M. (2017). Illumination change robustness in direct visual slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, pp. 4523–4530.

Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., & Humenberger, M. (2019). R2d2: Repeatable and reliable detector and descriptor. arXiv preprint arXiv:1906.06195.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *European conference on computer vision*, Springer, pp. 430–443.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *Proceedings of IEEE international conference on computer vision*, pp. 2564–2571.

Runz, M., Buffier, M., & Agapito, L. (2018). Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE international symposium on mixed and augmented reality (ISMAR)*, IEEE, pp. 10–20.

Sarlin, P.-E., Cadena, C., Siegwart, R., & Dymczyk, M. (2019). From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 12716–12725.

Shi, J., & Tomasi, C. (1993). *Good features to track*. Technical report, Cornell University.

Strasdat, H., Davison, A. J., Montiel, J. M., & Konolige, K. (2011). Double window optimisation for constant time visual slam. In *2011 International conference on computer vision*, IEEE, pp. 2352–2359.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the international conference on intelligent robot systems (IROS)*.

Sun, Y., Liu, M., & Meng, M.Q.-H. (2018). Motion removal for reliable rgb-d slam in dynamic environments. *Robotics and Autonomous Systems*.

Tang, C., & Tan, P. (2019). Ba-net: Dense bundle adjustment networks. In *2019 International conference on learning representations*.

Tang, J., Ericson, L., Folkesson, J., & Jensfelt, P. (2019). GCNv2: Efficient correspondence prediction for real-time slam. arXiv preprint arXiv:1902.11046.

Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6243–6252.

von Stumberg, L., Wenzel, P., Khan, Q., & Cremers, D. (2020). Gn-net: The gauss-newton loss for multi-weather relocalization. In *IEEE robotics and automation letters (RA-L)*, Vol. 5, No. 2, pp. 890–897.

Wang, S., Clark, R., Wen, H., & Trigoni, N. (2017). Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, pp. 2043–2050.

Yang, S., & Scherer, S. (2019). Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*.

Yang, N., Wang, R., Gao, X., & Cremers, D. (2018). Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4), 2878–2885.

Zhang, Z., Forster, C., & Scaramuzza, D. (2017). Active exposure control for robust visual odometry in HDR environments. In *2017 IEEE international conference on robotics and automation (ICRA)* IEEE, pp. 3894–3901.

Zhou, Y., Li, H., & Kneip, L. (2018). Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d-2-d edge alignment. *IEEE Transactions on Robotics*, 35(1), 184–199.

Huaiyang Huang received the B.Eng. degree in Mechatronics from Zhejiang University, Hangzhou, China, in 2018. He is now pursuing Ph.D. degree at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong.

Haoyang Ye received the B.Eng. in Automation from the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, HongKong University of Science and Technology, Hong Kong.

Yuxiang Sun received the Ph.D. degree from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2017, the master's degree from the University of Science and Technology of China, Hefei, China, in 2012, and the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009. He is now a research assistant professor at the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. Prior to that, he was a research associate at the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. Dr. Sun's research interests include autonomous driving, artificial intelligence, deep learning, mobile robots, etc.

Lujia Wang received the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, in 2015. She is now a research assistant professor at the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. From 2016–2021, she was an associate professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong. From 2015 to 2016, she was a research fellow with the School of Electrical Electronic Engineering, Nanyang Technological University, Singapore. Her current research interests include Cloud Robotics, Lifelong Federated Robotic Learning, Resource/Task Allocation for Robotic Systems, and Applications on Autonomous Driving, etc.

Ming Liu received the B.A. degree from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from ETH Zürich, Zürich, Switzerland, in 2013. He stayed one year at the University of Erlangen-Nuremberg, Erlangen, Germany, and FraunhoferInstitute IISB, Erlangen, as Visiting Scholar. His current research interests include dynamic environment modeling, 3-D mapping, machine learning, and visual control. Dr. Liu was the General Chair of ICVS 2017, the Program Chair of the IEEE RCAR 2016, and the Program Chair of the International Robotic Alliance Conference 2017.