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

# Monocular Visual Odometry using Learned Repeatability and Description

Huaiyang Huang[1], Haoyang Ye[1], Yuxiang Sun[1], Ming Liu[1], *Senior Member*, *IEEE*

***Abstract*— Robustness and accuracy for monocular visual odometry (VO) under challenging environments are widely concerned. In this paper, we present a monocular VO system leveraging learned repeatability and description. In a hybrid scheme, the camera pose is initially tracked on the predicted repeatability maps in a direct manner and then refined with the patch-wise 3D-2D association. The local feature parameterization and the adapted mapping module further boost different functionalities in the system. Extensive evaluations on challenging public datasets are performed. The competitive performance on camera pose estimation demonstrates the effectiveness of our method. Additional studies on the local reconstruction accuracy and running time exhibit that our system is capable of maintaining a robust and lightweight backend.**

## I. INTRODUCTION

Monocular visual odometry (VO) is a fundamental building block for robotic state estimation [1], providing critical information for high-level applications. With recent progress in this field, VO has been brought to maturity for various real-world deployments, e.g., VR/AR. Despite their success under moderate scenarios, diverse challenges [2], for instance, adverse illumination conditions [3], can lead the majority of current VO systems to fail. Therefore, the robustness and accuracy under challenging situations remain a problem demanding further investigation.

Emerging advances of deep learning (DL) provide an alternative perspective in resolving the aforementioned challenges. For examples, leveraging learning-based depth prediction [4], object recognition [5], or high-level semantics [6] benefits the visual state estimation in several aspects, varying from local reconstruction [7], scale recovery [8], [9] to dynamic environment handling [10]. Especially, in recent years, the exploration of learning local feature extraction and description is prevailing [11]. Following a *detect-and-describe* scheme to predict the repeatability and description within one forward pass, several works [12]–[14] demonstrate a superior efficiency against the traditional *detect-then-describe* pipelines, e.g., SIFT [15]. 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 VO systems. While some existing works generally recovering pose in an end-to-end manner [16], [17], limited the generalization ability for different scenarios, we consider a more practical approach to utilize the deep feature.

Motivated by these observations, we propose a monocular VO system powered by a learning-based frontend. Leveraging the repeatability maps 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 the pose estimation to mapping. 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. An overview of the proposed system is shown in Fig. 1. Our contributions are summarized as follows:

- A monocular visual odometry system leveraging learned description and repeatability.
- 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.

## II. RELATED WORKS

The state-of-the-art approaches for monocular VO can be categorized into two dominant classes, namely *indirect* and *direct* methods. Indirect, or feature-based methods [18]–[20], 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 [20] exploits the covisibility relationships to strengthen the map reuse and frame management, which well balances the accuracy and the computational demand. Direct methods [21]–[23], on the contrary, directly minimize the photometric error on input images to track camera poses. In particular, DSO [23] optimizes camera poses, sparse scene structure, and camera model parameters in a joint manner. To combine the advantages of both methods, Froster *et al*. propose SVO [24], which tracks the camera poses via sparse image alignment

[1]The authors are with the Robotics and Multi-Perception Laboratory in Robotics Institute at the Hong Kong University of Science and Technology, Hong Kong. (email: hhuangat@connect.ust.hk; hy.ye@connect.ust.hk; eeyxsun@ust.hk, sun.yuxiang@outlook.com; eelium@ust.hk).

This work was supported by the National Natural Science Foundation of China, under grant No. U1713211, and the Shenzhen Science, Technology and Innovation Commission (SZSTI) under grant JCYJ20160428154842603, the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, No. 21202816, awarded to Prof. Ming Liu.

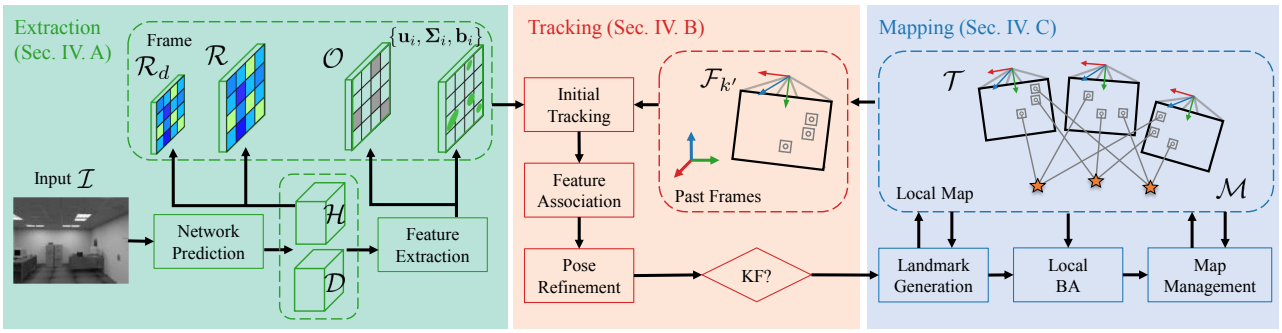


Fig. 1: Framework of the proposed system, where "KF" it the short of "Keyframe". Our system firstly decouples the network prediction as *repeatability* maps and 2d grid, and further extract sparse features parameterized as 2d saliency patches. Then, the tracking module optimizes the initial camera pose directly on the repeatability map. With feature associated with historical frames, the coarse pose estimation is then refined by motion-only BA. Finally, the tailored mapping module maintains the sparse feature map.

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 remains unsolved [1], [3], [25]. Potential solutions to these issues can be found in [26]–[28]. These works typically specialize in a particular problem (e.g. handling high dynamic range (HDR) environment), which would introduce certain overhead under common scenarios [26]. In contrast, our system is designed for a more general purpose.

Compared with traditional methods in local feature detection [29]–[31] and description [15], [32], [33], learning-based approaches [12]–[14], [34], [35] exhibit a competitive performance in both matching accuracy and efficiency. Several works propose to use deep feature into camera pose estimation. Tang *et al*. [36] 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 [37], 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 learned local feature with VO/VSLAM have been brought to our view [14], [38]. In [38], with labels from a VO backend, the original SuperPoint [12] is extended to predict the stability of local keypoints. In [14], 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 repeatability and description to alleviate the challenges in monocular VO.

## III. Notations

Throughout the paper, we denote the image collected at the $k$-th time as $\mathcal{I}_k$ 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 $\mathbf{p}_i \in \mathbb{R}^3$ in $\mathcal{F}_w$ to $\mathcal{F}_k$ using:

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

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, $^{c_k}\mathbf{p}_k$ denotes a 3D point in $\mathcal{F}_k$.

If a 2D pixel $\mathbf{u}^{\pi}_{i,k}$ is projected from a 3D landmark, it is right-superscripted by $\pi$. We use $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ to denote the projection function: $\mathbf{u}^{\pi}_{i,k} = \pi(^{c_k}\mathbf{p}_i)$, where $\mathbf{u}^{\pi}_{i,k}$ is the coordinate in the pixel coordinate. $\pi$ is defined as $\pi(^{c_k}\mathbf{p}_i) = \mathbf{K}^{c_k}\mathbf{p}_i$, 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 $\oplus : \mathfrak{se}(3) \times \mathbf{SE}(3) \to \mathbf{SE}(3)$ for the update of $\mathbf{T}_k$, which is denote as:

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

where $\boldsymbol{\xi}^{\wedge}$ is the skew-symmetric of $\boldsymbol{\xi}$.

## IV. Methodology

### A. Learned Repeatability and Description

Here we adopt SuperPoint [12] as the feature extraction front-end. Recall the pipeline of SuperPoint, it first encodes the input image $\mathcal{I} \in \mathbb{R}^{H\times W}$ 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, 8 in our system, and $H_c = H/C, W_c = W/C$. $\mathcal{H}'$ is then normalized by channel-wise softmax:

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

For the details we refer the readers to [12].

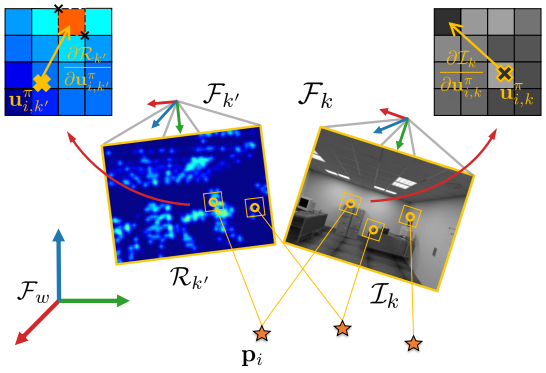


Fig. 2: A comparison of traditional direct tracking and the proposed. While photometry-based method seeks to minimize the intensity variance between landmarks and local observations on $\mathcal{I}_k$, our method constrains the current camera pose via requiring reprojection locations to reach the local minima of $\mathcal{R}_{k'}$.

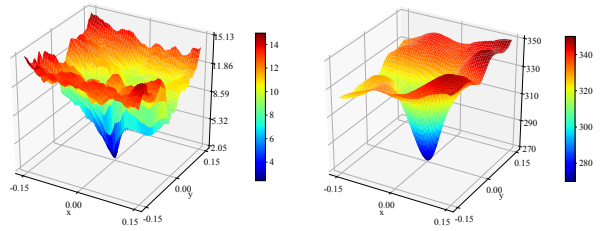

Fig. 3: Cost surface for camera pose tracking of *photometry*-based residual (left) and *repeatability*-based residual (right). For each plot, the x-y plane stands for different translational offsets to the ground-truth pose and the value on z-axis is the total cost with the corresponding transform.

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

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

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 Sec. IV-B. $\mathcal{R} \in \mathbb{R}^{H \times W}$ is the repeatability map with full-resolution and $s : \mathbb{R}^{H_c \times W_c \times C^2} \rightarrow \mathbb{R}^{H \times W}$ maps the volume to a 2D prediction. 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. In addition, we parameterize local features as 2D saliency patches, described in Sec. IV-B. 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 $\mathcal{D}'$.

### B. Camera Pose Tracking

*1) Initial Tracking on Repeatability:* A 3D landmark that can be observed cross different views is considered as a repeatable feature. Accordingly, its reprojection location is supposed to be the local peak on the repeatability map. Inspired by previous direct methods [23], [24], we introduce a *track-on-repeatability* approach that directly estimates camera poses based on $\mathcal{R}_d$ and $\mathcal{R}$. Fig. 2 illustrates our method and compares it with the traditional approach.

To estimate the current camera pose $\mathbf{T}_k$, for each point $\mathbf{p}_i$, the residual term based on the repeatability is defined as:

$$e_i^{\text{repeat}} = \mathcal{R}_{(\cdot)}(\pi_{(\cdot)}((\boldsymbol{\xi}_k \oplus \mathbf{T}_k)\mathbf{p}_i), \tag{5}$$

where $\mathcal{R}_{(\cdot)}$ stands for repeatability maps with different resolutions ($\mathcal{R}_d$ and $\mathcal{R}$). $\pi_{(\cdot)}$ is the camera projection function according to the resolution of repeatability map. The Jacobian of the residual term $e_i^{\text{repeat}}$ is derived as:

$$\begin{aligned}\mathbf{J}_i &= \mathbf{J}_{\text{repeat}} \cdot \mathbf{J}_{\text{proj}} \cdot \mathbf{J}_{\text{pose}} \\ &= \frac{\partial \mathcal{R}_{(\cdot)}(\mathbf{u}_{i,k}^{\pi})}{\partial \mathbf{u}_{i,k}^{\pi}} \cdot \frac{\partial \mathbf{u}_{i,k}^{\pi}}{\partial \mathbf{p}_i^c} \cdot \frac{\partial \left((\boldsymbol{\xi}_k \oplus \mathbf{T}_k)\,\mathbf{p}_i\right)}{\partial \boldsymbol{\xi}_k},\end{aligned} \tag{6}$$

where similar to direct tracking methods, $\mathbf{J}_{\text{repeat}}$ is the gradient of the corresponding repeatability map, which is evaluated at the projected pixel $\mathbf{u}_{i,k}^{\pi}$ and is calculated by bilinear interpolation at each iteration. $\mathbf{J}_{\text{proj}}$ and $\mathbf{J}_{\text{pose}}$ 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 $\boldsymbol{\xi}_k$, respectively.

$\boldsymbol{\xi}_k$ at each optimization step is solved by minimizing the overall energy function:

$$\hat{\boldsymbol{\xi}}_k = \arg\min_{\boldsymbol{\xi}_k} \sum_{i \in \mathcal{P}_k} \|\mathcal{R}_{(\cdot)}(\pi_{(\cdot)}(\boldsymbol{\xi}_k \oplus \mathbf{T}_k)\mathbf{p}_i)\|_{\gamma}. \tag{7}$$

where $\mathcal{P}_k$ is initially assigned as the set of landmarks tracked by the previous frame $\mathcal{F}_{k-1}$. If $|\mathcal{P}_k|$ is too small, we insert points visible by other keyframes sharing high covisibility with $\mathcal{F}_{k-1}$ into $\mathcal{P}_k$ to guarantee $|\mathcal{P}_k|$ is larger than a minimum requirement. And the pose $\mathbf{T}_k$ is updated by $\mathbf{T}_k \leftarrow \boldsymbol{\xi}_k \oplus \mathbf{T}_k$ iteratively until convergence or the maximum number of iterations is reached. We solve the above problem via standard Levenberg-Marquardt method.

Fig. 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. The detailed explanation and discussion in [39].

*2) Pose Refinement:* To refine the pose solved in the initial tracking, we first associate 3D landmarks with local features in patch-level. If the optimization converges, the reprojection location $\mathbf{u}_{i,k}^{\pi}$ of an inlier $\mathbf{p}_i$ is supposed to have sub-patch accuracy. In another word, the local correspondence of $\mathbf{p}_i$ belongs to $\mathcal{V}^{\text{adj}}(\mathbf{u}_{i,k}^{\pi})$, the set of four adjacent cells. We check the gird 2d along with descriptor distance to find the best

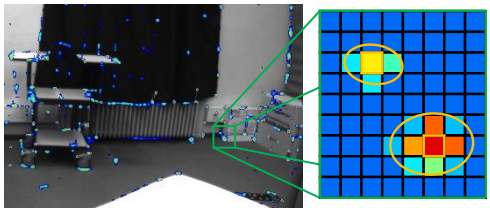

Fig. 4: Local feature extraction and parameterization. The input grayscale image overlaied by the responses from the predicted repeatability map is shown, with an example zoom-in view of local features. We parameterize each feature on the image as a 2D saliency patch with the local peak □ and covariance ◯.

local association $\tilde{\mathbf{u}}_{i,k}$. After associating landmarks with the local observations, the current camera pose $\mathbf{T}_k$ is refined by minimizing the reprojection error. For $\mathbf{p}_i$, the error function is defined as:

$$\mathbf{e}_{i,k}^{\text{repro}} = \pi\left(\mathbf{R}_k\mathbf{p}_i + \mathbf{t}_k\right) - \tilde{\mathbf{u}}_{i,k}. \tag{8}$$

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

$$E^{\text{repro}} = \sum_{i\in\mathcal{P}_k} \| \left(\mathbf{e}_{i,k}^{\text{repro}}\right)^T \mathbf{\Sigma}_{i,k}^{-1} \mathbf{e}_{i,k}^{\text{repro}} \|_\gamma, \tag{9}$$

where $\mathbf{\Sigma}_{i,k}$ is the covariance of 2D feature location $\tilde{\mathbf{u}}_{i,k}$ for weighting different features' contribution to the optimization. In [20], [24], $\mathbf{\Sigma}$ is dependent on the pyramid level on which the corresponding feature is detected. Compared to traditional feature extraction techniques, CNNs have a larger receptive field and is capable of generating more consistent local features. Therefore, we model a 2D feature as a local saliency patch with mean and covariance. To approximate the covariance of feature, as illustrated in Fig. 4, we extract local saliency patches to represent the uncertainty in the prediction. $\mathbf{\Sigma}$ is derived as:

$$\begin{aligned}\mathbf{\Sigma} &= E\left[\left(\mathbf{u}-\mathbf{u}_{\text{peak}}\right)\left(\mathbf{u}-\mathbf{u}_{\text{peak}}\right)^T\right] \\ &= \sum_{i\in\mathcal{V}^{\text{p}}} \frac{\mathcal{R}(\mathbf{u}_i)}{\sum_{k\in\mathcal{V}^{\text{p}}}\mathcal{R}(\mathbf{u}_k)} \left(\mathbf{u}_i-\mathbf{u}_{\text{peak}}\right)\left(\mathbf{u}_i-\mathbf{u}_{\text{peak}}\right)^T,\end{aligned} \tag{10}$$

where $\mathbf{u}_{\text{peak}}$ is the local peaky pixels extracted from repeatability map. $\mathcal{V}^p$ is the set of pixels adjacent to $\mathbf{u}_{\text{peak}}$ and with positive responses.

To avoid computational overhead, the direct tracking is performed on $\mathcal{R}_d$ for most of the time. If in the refinement step the matches are not enough, we again track the pose on $\mathcal{R}$, which has a larger resolution. In practice, this situation seldom happens and we notice that the coarse tracking generally takes less than 10 steps for convergence.

### C. Mapping

*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-SLAM [20] leverages bag-of-words (BoW) [40] to accelerate feature matching between keyframes for the landmark generation. However, as we extract the most representative features and suppress the number of redundancies via NMS, matching with BoW can significantly reduce the candidate matches and thus the number of landmarks for ego-motion estimation, which degrades both the robustness and the accuracy. 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.

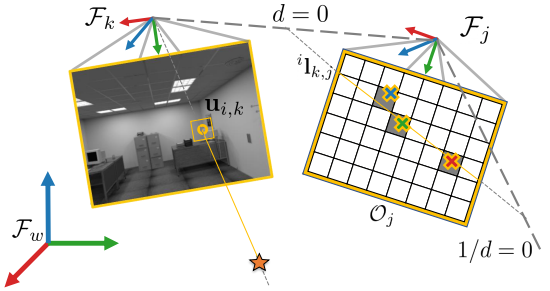


Fig. 5: Landmark generation. The landmarks are generated by searching the 2d grid of patches along the epipolar line. Outliers are rejected by the epipolar constraint (×) or the Euclidean distance of descriptors (×). The final inlier (×) is triangulated to retrieve the 3D position.

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:

$$({}^i\mathbf{l}_{k,j})^T = \mathbf{u}_{i,k}^T\mathbf{K}^{-T}\mathbf{t}_{k,j}^{\wedge}\mathbf{R}_{k,j}\mathbf{K}^{-1},$$

where $\mathbf{K}$ is the projection matrix. $\mathbf{T}_{k,j} = [\mathbf{R}_{k,j}, \mathbf{t}_{k,j}]$ is relative transform from $\mathcal{F}_j$ to $\mathcal{F}_k$. As illustrated in Fig. 5, 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 weighted 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 position are recovered via mid-point triangulation. For rejecting outliers in triangulation, we follow [20] to further verify the sign of depth and the reprojection error to both keyframes.

TABLE I: Translational RMSE (cm) and tracking success rate (%) on the New Tsukuba dataset. The tracking success is defined as (#tracked frame/#frame in the sequence).

| Sequence Name | Ours | DSO | ORB-SLAM w/o loop |
|---|---|---|---|
| fluorescent | **10.9 ± 5.7**<br>87.6 ± 0.3 | 59.0 ± 39.0<br>98.2 ± 0.0 | (18.2 ± 15.3)<br>88.4 ± 11.6 |
| daylight | **9.4 ± 4.9**<br>96.2 ± 0.2 | 50.0 ± 29.7<br>98.1 ± 0.0 | 14.7 ± 9.7<br>82.4 ± 16.9 |
| lamps | **9.3 ± 4.6**<br>94.5 ± 7.1 | ×<br>44.5 ± 37.4 | ×<br>0.0 ± 0.0 |
| flashlight | **18.3 ± 10.3**<br>94.4 ± 0.1 | ×<br>58.4 ± 10.9 | ×<br>0.0 ± 0.0 |

TABLE II: Translational RMSE (cm) on the EuRoC dataset.

| Seq | | Ours | DSO | ORB-SLAM w/o loop |
|---|---|---|---|---|
| easy | MH01 | **1.63 ± 0.86** | 5.73 ± 2.28 | 1.77 ± 0.91 |
| | MH02 | **1.46 ± 0.77** | 4.53 ± 2.71 | 1.72 ± 0.81 |
| | MH03 | 3.20 ± 1.53 | 20.89 ± 10.23 | **3.19 ± 1.52** |
| | V101 | 3.34 ± 4.34 | 13.52 ± 8.72 | **3.28 ± 1.20** |
| | V201 | **1.85 ± 0.83** | 4.70 ± 2.41 | 2.59 ± 1.34 |
| | V202 | **4.73 ± 2.49** | 12.31 ± 6.37 | (5.08 ± 2.95) |
| hard | MH04 | **8.59 ± 3.37** | 20.03 ± 8.96 | 9.78 ± 4.05 |
| | MH05 | **4.34 ± 2.09** | 10.47 ± 3.78 | (12.67 ± 5.86) |
| | V102 | **23.82 ± 9.38** | 32.83 ± 27.41 | (14.06 ± 5.27 ) |
| | V103 | (17.17 ± 10.71) | **94.15 ± 39.78** | × |
| | V203 | × | × | × |

*2) Backend Optimization:* Batched BA is utilized for backend optimization and map management. The variables in the local map to optimize are consisted of updates to keyframe poses $\mathcal{T} = [\boldsymbol{\xi}_{k_0}, \boldsymbol{\xi}_{k_1}, ..., \boldsymbol{\xi}_{k_n}]$ and positions of landmarks $\mathcal{M} = [\mathbf{p}_{i_0}, \mathbf{p}_{i_1}, ..., \mathbf{p}_{i_m}]$. We denote the full state vector as $\mathcal{X} = [\mathcal{T}, \mathcal{M}]$, which is solved by:

$$\mathcal{X} = \arg\min_{\mathcal{X}} \sum_{\mathbf{T}_k} \sum_{\mathbf{p}_i} \text{obs}(i,k) \| (\mathbf{e}^{\text{repro}}_{i,k})^T \boldsymbol{\Sigma}^{-1}_{i,k} \mathbf{e}^{\text{repro}}_{i,k} \|_\gamma, \quad (11)$$

where $\text{obs}(i,k) = 1$ if $\mathbf{p}_i$ is observed by $\mathcal{F}_k$, $\text{obs}(i,k) = 0$ otherwise. Note that, as in ORB-SLAM [20], 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 adapted from [20]. Briefly, it culls redundant keyframes, removes outliers in the local BA and fuses the landmarks.

## V. Experimental Results

### A. Evaluation on Trajectory Estimation

We evaluate our system on two different datasets, the New Tsukuba [41] and the EuRoC Mav dataset [42]. We compare our method to both the state-of-the-art indirect (ORB-SLAM [20]) and direct (DSO [23]) VO algorithms. GCN-SLAM [14], 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. In addition, for a fair comparison, the loop-closure module of ORB-SLAM is manually turned off.

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) [43] is used. We run different algorithms on each sequence 10 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 $(\cdot)$, while failure for all runs is marked as $\times$. The results on both datasets are reported in Table. I and Table. II, respectively.

*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-SLAM 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.

Table. I 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 [23].

*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. 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

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

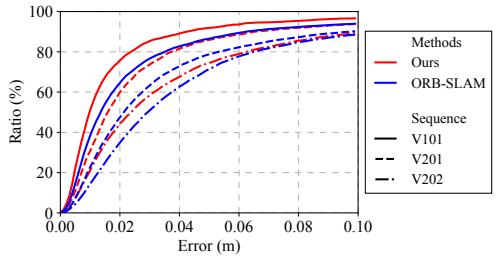


Fig. 6: Evaluation on the local reconstructions. The higher the percentage of points towards a zero distance the better.

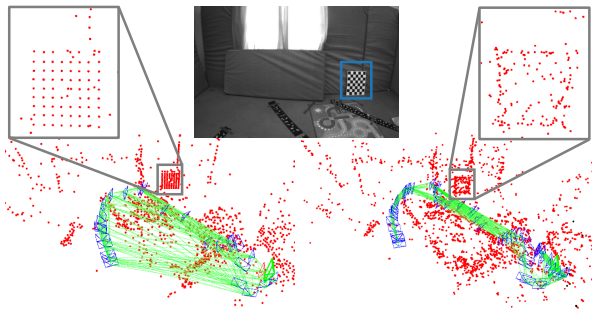

Fig. 7: Qualitative evaluation of covisibility graph (line) with 10% common observations and sparse structure reconstruction (point) in an indoor environment (top centre). The zoom-in views (□) of a checkboard (□) are shown for comparing the quality of sparse maps.

challenges for monocular VO, especially indirect methods.

As shown in Table. II, 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.

### B. Evaluations on Local Reconstruction

To evaluate the reconstruction accuracy, the sparse point-clouds generated by monocular VOs 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 [44]. Three sequences V101, V201, and V202 of EuRoC that provide dense ground-truth structure are selected for the evaluation.

As shown in Fig. 6, 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-SLAM, the local structure is more accurate than ORB-SLAM, indicating fewer outliers in our system. Fig. 7 provides a qualitative comparison of the mapping performance. 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 [23]. ORB-SLAM 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.

TABLE III: Runtime analysis on EuRoC V201.

| | Module | Ours | ORB-SLAM |
|---|---|---|---|
| Tracking | Feature Extraction (ms) | 17.8 ± 1.2 | **11.5 ± 1.3** |
| | Coarse tracking (ms) | **1.6 ± 0.2** | 2.2 ± 0.7 |
| | Pose Refinement (ms) | **3.5 ± 0.6** | 6.1 ± 1.1 |
| | Total (ms) | 23.4 ± 1.7 | **20.5 ± 3.9** |
| Mapping | Map Management (ms) | 63.0 ± 16.9 | **55.8 ± 8.4** |
| | Local BA (ms) | **58.1 ± 52.8** | 131.1 ± 167.3 |
| | Total (ms) | **121.0 ± 67.5** | 186.9 ± 179.5 |
| | #Keyframes | **139** | 458 |
| | Accumulated (s) | **17.6** | 85.6 |

### C. Runtime Analysis

To further compare the efficiency against ORB-SLAM, we evaluate the detailed runtime performance of modules with similar functionality of both systems. The evaluation results on EuRoC V201 are shown in Table. III.

Although matching float descriptors is more time-consuming than binary ones, our method still improves the tracking time by 40%, comparing the total tracking time in Table. III. This is because we only extract the most representative keypoints and estimate the coarse pose in advance of final pose estimation. It is also noticeable that our system maintains a mapping backend much more lightweight than ORB-SLAM (with only 20% computation time), thanks to the better capability of cross-view data association.

Admittedly, the current implementation introduces certain overhead as shown in Table. III: 1) Feature extraction part. An over 80% improvement of inference time via knowledge distillation is demonstrated in [35], which can be regarded as a potential solution. 2) Descriptor distance computation. Although brute-force search is avoided in our system, overhead is still observed, e.g., in the map management module.

## VI. Conclusions and Future Work

In this paper, we presented a monocular VO system leveraging learned description and repeatability. Different from previous work, we focused on tightly coupling the learning-based frontend with indirect multiview geometry to fully exploit the network predictions. We 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. The adapted mapping module maintained a lightweight and consistent backend. 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 supervising the learning frontend in an end-to-end manner.

## REFERENCES


[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable rgb-d slam in dynamic environments," *Robotics and Autonomous Systems*, 2018.

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

[4] R. Mahjourian, M. Wicke, and A. Angelova, "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*, 2018, pp. 5667–5675.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

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

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

[9] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, 2019.

[10] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, pp. 10–20.

[11] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182.

[12] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

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

[14] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient correspondence prediction for real-time slam," *arXiv preprint arXiv:1902.11046*, 2019.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

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

[18] G. Klein and D. Murray, "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, 2007, pp. 1–10.

[19] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2352–2359.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

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

[23] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

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

[25] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2878–2885, 2018.

[26] H. Alismail, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2016.

[27] Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in hdr environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3894–3901.

[28] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim, "Robust visual localization in changing lighting conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5447–5452.

[29] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[30] J. Shi and C. Tomasi, "Good features to track," Cornell University, Tech. Rep., 1993.

[31] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. Int. Conf. Computer Vision*, Nov. 2011, pp. 2564–2571.

[34] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," *arXiv preprint arXiv:1906.06195*, 2019.

[35] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[36] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment networks," in *2019 International Conference on Learning Representations*.

[37] L. von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "Gn-net: The gauss-newton loss for multi-weather relocalization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 890–897, 2020.

[38] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Self-improving visual odometry," *arXiv preprint arXiv:1812.03245*, 2018.

[39] Y. S. Huaiyang Huang, Haoyang Ye and M. Liu, "Monocular visual odometry using learned repeatability and description: Supplementary materials," [Online]. https://sites.google.com/view/rdvo/.

[40] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[41] S. Martull, M. Peris, and K. Fukui, "Realistic cg stereo image dataset with ground truth disparity maps," in *ICPR workshop TrakMark2012*, vol. 111, no. 430, 2012, pp. 117–118.

[42] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[43] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[44] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, "C-blox: A scalable and consistent tsdf-based dense mapping approach," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 995–1002.