TRVO: A Robust Visual Odometry with Deep Features

Yuhang Gao[1], Long Zhao [*1]

[*1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
*E-mail: flylong@buaa.edu.cn

Abstract. Accurate localization is crucial for visual SLAM systems. However, most visual SLAM systems use traditional hand-crafted local features to find matches in two images, which are less stable in scenes with textures-less, motion blur or repetitive patterns, and cannot achieve the goal of lifelong SLAM. In this paper, we propose TRVO, a visual odometer that uses deep learning for feature matching. The deep learning network adopts the structure of CNN and Transformer, which can produce high-quality dense matches for a pair of images in an end to end form even in indistinctive scenes, where low-texture regions or repetitive patterns occupy most areas in the field of view. After the matching point pairs are obtained, the camera pose is solved in an optimized way by minimizing the reprojection error of the feature points. Experiments based on multiple dataset and real environments show that TRVO has higher relative positioning accuracy and robustness compared with the current mainstream visual SLAM systems.

Keywords: Feature Mapping, Deep Learning, Visual Odometry, Transformer

1. INTRODUCTION

The technology of simultaneous localization and mapping (SLAM) has been greatly developed in recent years. As vision sensors can provide rich environmental texture information, there are more and more researches on vision SLAM algorithm in the community. As a widely used method, feature matching plays a vital role in the application of robot localization and navigation. However, most of these feature matching operation heavily depends on the descriptors of the keypoints. These keypoints are not stable enough due to various factors such as low-texture, illumination variation, viewpoint changes, and motion blur, and fail to complete the positioning task.

The importance of keypoint description motivated its extensive research that has resulted in the development of many description techniques. Early studies focused on hand-crafted methods, such as SIFT [2], SURF [3], and ORB [4]. They still play a substantial role in computer vision task today. However, these traditional feature extraction and matching methods are easily affected by extreme lighting changes, motion blur, repetition and texture-less scenes. With the development of parallel computing capability of computer, the feature extraction and matching method based on deep learning has gradually replaced the hand-crafted ones to become the mainstream of research and application in most computer vision tasks for it is more robust in challenging scenes.

Recent works attempt to solve the problem of traditional hand-crafted methods by establishing dense matching in pixel level. Matching pairs are directly selected from dense matching results with high confidence, which omits the process of feature detection. However, the receptive field of dense features extracted by convolution is limited, and it cannot distinguish the regions with high similarity. Therefore, the author of LoFTR [5] integrated Transformer [6] into feature matching and proposed a novel local feature matching method without detection process. Self and cross attention layers are used to process the dense local features extracted from convolution backbone. Compared to detector-based baseline methods, LoFTR can generate high-quality matches even in indistinctive regions with low-texture, motion blur, or repetitive patterns.

In this paper, we propose TRVO, a visual odometry with Transformer local feature. We introduce the LoFTR method into the visual odometry, and get more accurate pose estimation results as LoFTR can produce high-quality feature matches between two images. Firstly, two consecutive frames in a continuous image stream are put into the LoFTR network. Then the visual odometry system minimizes the reprojection error through nonlinear optimization, and iteratively obtains the optimal pose estimation between two frames. The remainder of this paper is organized as follows: the second part introduces the related research work of visual odometry. The third part introduces the working principle of LoFTR network and the derivation of reprojection error formula. In the fourth part, TRVO is extensively evaluated on dataset and real environments.

2. RELATED WORK

In this section, we will review the existing work related to visual SLAM and image feature matching.

2.1. Visual SLAM System

The general flow of the visual SLAM is shown in **Fig. 1**, it takes camera as the main sensor to establish the spatial relationship between camera and surrounding environment. For the sequence images or videos captured by the camera with a given internal parameter, the visual SLAM system calculates the camera motion trajectory and establishes the environment map by estimating the camera motion between the key frames. The visual front end is also known as the visual odometry, which uses the image data collected by the sensor to estimate the position and orientation relationship between key frames.

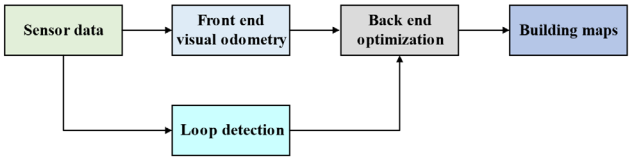


Fig. 1 Visual SLAM system flow.

The visual odometry can be divided into two categories: feature-based method and direct method. The visual front-end based on feature points has been considered as the mainstream visual odometry method for a long time. MonoSLAM [7] is the first real-time monocular vision SLAM system, which uses extended Kalman filter as the back end, tracks the very sparse feature points in the front end, and updates its mean and covariance by taking the current state of the camera and all landmarks as state variables. PTAM [8] proposes to parallelize the tracking and mapping process, and incorporates nonlinear optimization into SLAM system for the first time. Meanwhile, PTAM introduces key frame mechanism to reduce the computational complexity of optimization. ORB-SLAM [9-11] is a representative SLAM system of feature-based method. The whole system revolves around the calculation of ORB features, it can detect loops in large-scale motion and relocate if the tracking is lost. The VINS [12] system adopts the fusion SLAM scheme of vision-inertial, and the visual odometry adopts the sparse direct method, that is, only the feature points are calculated, not the descriptors, and the optical flow method is used to track the motion of the feature points, which can save the time of descriptor calculating and matching.

With the development of the application of deep learning in computer vision tasks, there have been some researches on visual slam utilizing deep learning for keypoints detect and match in recent years. For example, GCNv2 [13] integrates deep local features into ORB-SLAM2 framework. DXSLAM [14] is similar to GCNv2, which is also based on ORB-SLAM2 and uses HF-Net [15] to generate SuperPoint-like [16] feature points and descriptors. DXSLAM is with higher robustness in indistinctive scenes such as low-texture and illumination change compared with slam system which uses hand-crafted keypoints and descriptors.

2.2. Feature extraction and matching for images

In the numerous traditional works of feature detection and matching, SIFT and ORB are arguably the most successful methods, they are widely used in many 3D computer vision tasks. However, with the development of deep learning, the feature extraction and matching method based on learning has a better performance in dealing with the challenges such as extreme illumination and viewpoint change. MagicPoint [17] is the first successful local feature detection method based on learning. Then SuperPoint proposed a self-supervised training method through the improved homography matrix based on MagicPoint. The above two methods are to find the matching point pairs between the extracted keypoints by the way of nearest neighbor search. SuperGlue [18] proposed a local feature matching method based on learning. It takes two sets of keypoints with descriptors as input, and uses graph neural network (GNN) to learn the matches of the two sets of keypoints. However, as a detector dependent method, it cannot generate repeatable features in indistinctive regions.

Recently, researchers have proposed some local feature matching methods without detector, such as LoFTR and Pach2Pix [19]. They remove the feature detection step, and perform feature detection, descriptor calculation and feature matching in a single network. It is noteworthy that the matches of two images generated by Pach2Pix network need to be further selected by RANSAC [20] method. While LoFTR introduces the Transformer structure, which will be described in detail in the third part, thus can obtain the global information of the image through self and cross attention layers. Dense matching results can be obtained even in regions with low-texture using LoFTR. However, it is difficult for detector dependent methods to generate repeatable keypoints in such regions with low distinguishability. Therefore, in our TRVO system, we use the matching result generated by LoFTR as the input of the visual odometry, and then minimize the reprojection error by nonlinear optimization method to get the pose estimation results of each frame.

3. METHOD

3.1. System Overview

The visual odometry system proposed in this paper is shown in **Fig. 2**. The framework is similar to the front end of ORB-SLAM. The system processes two adjacent frames collected by the camera sensor each time. That is to say, the last frame and the current frame are passed through the LoFTR network to get the feature matching result. The network will score each pair of matching points, select the matching point pairs whose scores are greater than our preset threshold, and calculate the coordinates of these points in 3D space. Then, according to the pose results of the last frame obtained in the previous optimization, the spatial coordinate positions of these matching point pairs relative to the origin of the world coordinate system (the spatial position of the camera when taking the first frame image) are generated,

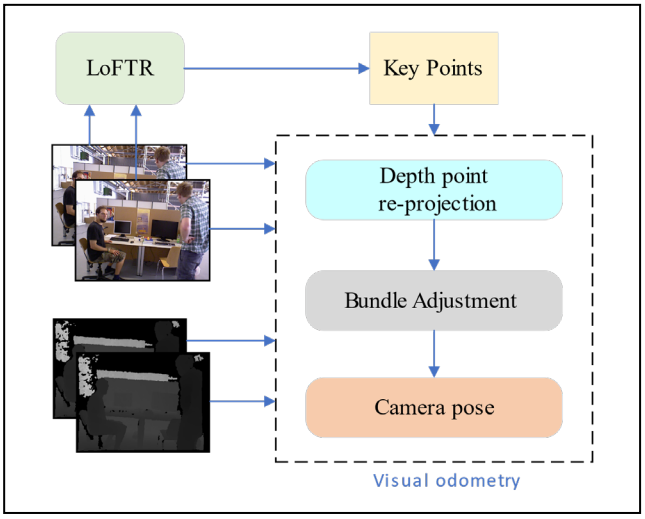


Fig. 2 The framework of the proposed visual odometry system. The pipeline is similar to the front end of ORB-SLAM, with deep features incorporated into the system.

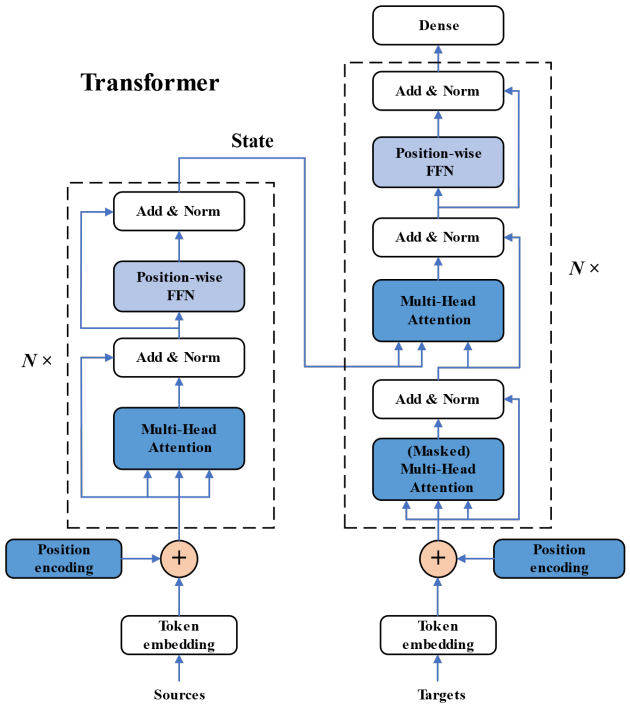


Fig. 3 The architecture of Transformer

and these positions are re-projected to the current frame. The relative pose relationship between the last frame and the current frame is obtained by minimizing the reprojection error in an optimized way. If the whole process continues, a complete visual odometry can be obtained.

3.2. Transformer and LoFTR

Transformer is a classic natural language processing (NLP) model proposed by Google in 2017. It uses self and cross attention mechanism to enable the model to train in parallel and obtain global information. Recently, Transformer has attracted more and more attention in computer vision tasks, such as image segmentation, object detection, semantic segmentation and other tasks. The architecture of Transformer is shown in **Fig. 3**, which is mainly composed of encoder and decoder. Encoder is composed of N identical layers, and each layer is composed of two sub layers, namely multi head self- attention and fully connected feed forward network. The structure of decoder is similar to that of encoder, but a multi attention sub layer is added. In addition to encoder and decoder, it is necessary to encode the raw data in the preprocessing period, so that the network can provide global receptive. For more details, please refer to [6].

The LoFTR network use self and cross attention layers in Transformer to obtain feature descriptors that are conditioned on both images. As shown in **Fig. 4** [5], the LoFTR network consists of four parts: for the first step, the coarse-level and fine-level feature maps are generated for the input image pair through a CNN backbone. Then the coarse feature maps is flattened to one-dimensional vector, and positional encoding is added. The added features are then processed by the Local Feature Transformer module, which has Nc self-attention and cross-attention layers. Next, the transformed features are matched by a differentiable matching layer. In this case, a differentiable matching layer confidence matrix Pc can be obtained, the a coarse-level match prediction Mc are selected according to the confidence threshold and mutual-nearest-neighbor criteria in Pc. Finally, for each coarse-level match prediction, a local window with size w × w is generated on the corresponding position of the fine-level feature maps. The Features in this window will be processed again by the Local Feature Transformer module, yielding a sub-pixel level match prediction.

3.3. Nonlinear optimization of reprojection error

In the camera pose solving problem, if only the matching point pairs in two images are known, that is, 2D-2D point pairs, it is necessary to use at least eight pairs of point to calculate the relative motion of camera by epipolar geometry method, and there are some problems such as initialization, pure rotation and scale. If the 3D position of one of the feature points in two images is known, at least three pairs of point are needed to estimate the camera motion. Therefore, for convenience, we will use RGB-D camera as the sensor for system design and experiment.

If the 3D position of n feature points is known, a nonlinear least square problem can be constructed directly to obtain the relative motion of the camera between two images by minimizing the reprojection error. This linear optimization problem can be easily solved by g2o [21], Ceres and other optimization libraries. Considering a set P contains n 3D points and their projection p in one image, our goal is to calculate the pose T of the camera, which is composed of rotation R and displacement t. Suppose that the spatial coordinate of a 3D point is $\boldsymbol{P}_i = [X_i, Y_i, Z_i]^T$ and the pixel coordinate of its projection in the image is $\boldsymbol{u}_i = [u_i, v_i]^T$,

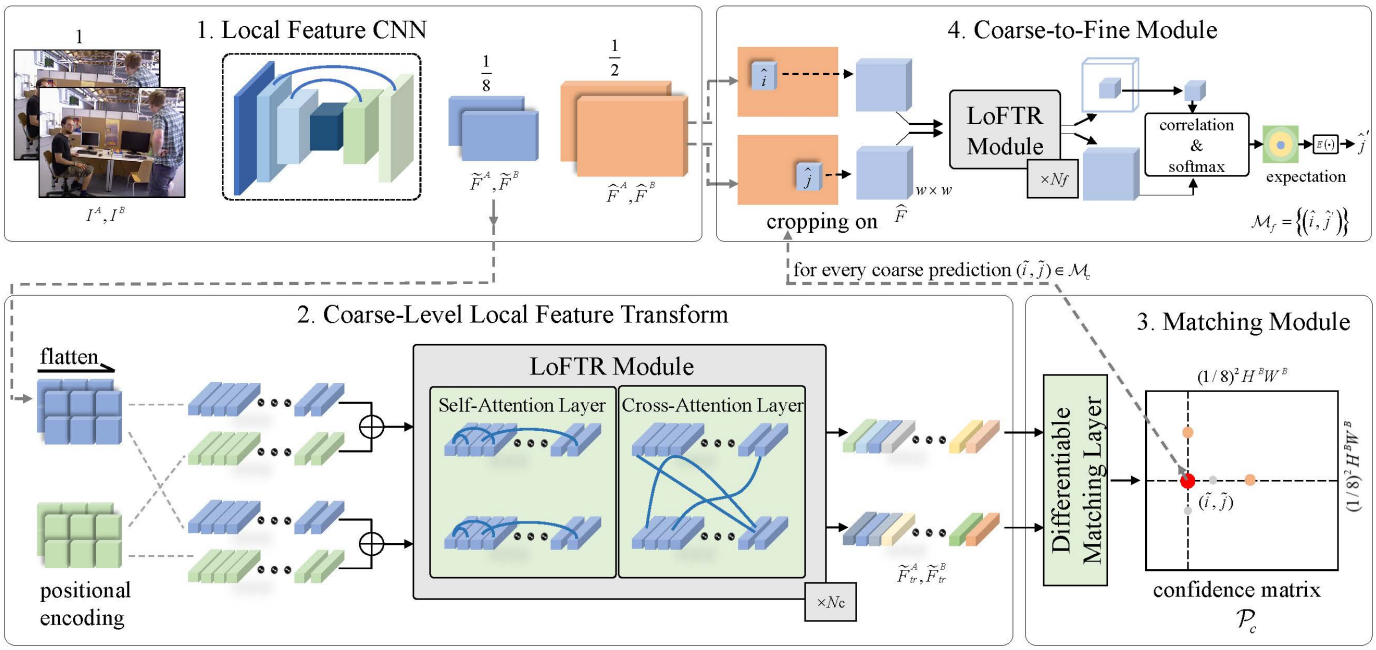


Fig. 4 LoFTR pipeline

then the relationship between the pixel position and the 3D point position satisfies the following equation

$$Z_i \boldsymbol{P}_{uv} = Z_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} (\boldsymbol{RP}_i + \boldsymbol{t}) = \boldsymbol{K}(\boldsymbol{RP}_i + \boldsymbol{t}) = \boldsymbol{KTP}_i \tag{1}$$

where K is the intrinsic parameter matrix of the camera, and $\boldsymbol{P}_{uv}$ is the homogeneous coordinate representation of $\boldsymbol{u}_i$. At this time, because the pose of the camera is unknown, there is an error between the two sides of (1). We sum all the errors of n feature points and construct the least square problem (2). We can find the optimal pose estimation when the error is minimum.

$$\boldsymbol{T}^* = \arg\min_{T} \frac{1}{2} \sum_{i=1}^{n} \left\| \boldsymbol{u}_i - \frac{1}{Z_i} \boldsymbol{KTP}_i \right\|_2^2 \tag{2}$$

After the optimization problem is constructed, Gauss-Newton, Levenburg-Marquardt and other optimization algorithms can be used to solve the pose, but the premise is to know the derivative of each error term with respect to the optimization variable. Now, we transform $\boldsymbol{P}_i$ to camera coordinate to get p'

$$\boldsymbol{P}'_i = \left(\exp\left(\boldsymbol{\xi}^\wedge \right) \boldsymbol{P}_i \right)_{1:3} = \left[X'_i, Y'_i, Z'_i \right]^T \tag{3}$$

where $\boldsymbol{\xi}$ is the lie algebra representation of the camera pose, the coordinates in the formula are homogeneous, and we take the first three-dimensional after transformation. According to the projection model of the camera, the following formula can be obtained

$$\begin{cases} u = f_x \dfrac{X'_i}{Z'_i} + c_x \\ v = f_y \dfrac{Y'_i}{Z'_i} + c_y \end{cases} \tag{4}$$

When calculating the errors, we make the difference between the coordinate value calculated by the above formula and the actual pixel value measured in the image. According to the chain rule we have

$$\frac{\partial \boldsymbol{e}}{\partial \delta \boldsymbol{\xi}} = \lim_{\delta \xi \to 0} \frac{e(\delta \boldsymbol{\xi} \oplus \boldsymbol{\xi}) - e(\boldsymbol{\xi})}{\delta \boldsymbol{\xi}} = \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P}'} \frac{\partial \boldsymbol{P}'}{\partial \delta \boldsymbol{\xi}} \tag{5}$$

and

$$\frac{\partial e}{\partial \boldsymbol{P}'} = -\begin{bmatrix} \dfrac{\partial u}{\partial X'_i} & \dfrac{\partial u}{\partial Y'_i} & \dfrac{\partial u}{\partial Z'_i} \\ \dfrac{\partial v}{\partial X'_i} & \dfrac{\partial v}{\partial Y'_i} & \dfrac{\partial v}{\partial Z'_i} \end{bmatrix} = -\begin{bmatrix} \dfrac{f_x}{Z'_i} & 0 & -\dfrac{f_x X'_i}{Z'^2} \\ 0 & \dfrac{f_y}{Z'_i} & -\dfrac{f_y Y'}{Z'^2} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{P}'}{\partial \delta \boldsymbol{\xi}} = \left[\boldsymbol{I}, -\boldsymbol{P}'^\wedge \right]$$

Then we have the Jacobian matrix by multiplying the above two terms

$$\frac{\partial e}{\partial \delta \boldsymbol{\xi}} = -\begin{bmatrix} \dfrac{f_x}{Z'_i} & 0 & -\dfrac{f_x X'_i}{Z'^2_i} & -\dfrac{f_x X'_i Y'_i}{Z'^2_i} & f_x + \dfrac{f_x X_i^2}{Z'^2_i} & -\dfrac{f_x Y'_i}{Z'_i} \\ 0 & \dfrac{f_y}{Z'_i} & -\dfrac{f_y Y'_i}{Z'^2_i} & -f_y - \dfrac{f_y Y_i'^2}{Z'^2_i} & \dfrac{f_y X'_i Y'_i}{Z'^2_i} & \dfrac{f_y X'_i}{Z'_i} \end{bmatrix}$$

4. EVALUTION

In this section, we perform a series of experiments to evaluate our proposed TRVO system. As mentioned in (3.3), we use Intel RealSense D435i camera (an RGB-D camera with IMU) and RGB-D dataset for convenience. We compare the positioning accuracy of TRVO with open source solutions such as ORB-SLAM2 and DXSLAM. All the methods are compiled and run on a laptop computer with i7-7700HQ processor and GTX 1050 Ti graphics card in Ubuntu Linux system.

4.1. Feature matching

In the first experiment, we compare the feature matching performance of ORB feature, HF-Net and LoFTR network. The images were taken from the TUM RGB-D dataset, Aachen Day-Night dataset [22] dataset and corridor environment captured by Intel RealSense D435i camera.

Experimental Setup. For ORB feature and HF-Net, given a pair of images, keypoints and descriptors of each image are generated, then we can get matches according to Hamming distance between descriptors. After that, RANSAC algorithm is used to further filter the correct matching results. For LoFTR network, we use the confidence scores produced by the network to filter out outliers. The confidence represents a trade-off between quality and quantity of the matches. In this experiment, the confidence score threshold was set to $c = 0.0/0.5/0.8$.

Results. We show qualitative results in **Fig. 5**. The ORB features are sparse, and there are many mismatches; The features learned by HF-Net are also sparse, but with higher matching accuracy; LoFTR network generates dense match pairs, we can select high-quality matching points according to high confidence. Compared with ORB feature and HF-Net, LoFTR performed better under low-texture environment. The results show that LoFTR is a better choice for indoor environment with low-texture regions.

4.2. Localization on TUM RGB-D dataset

To evaluate the positioning accuracy of our proposed visual odometry, we carry out visual positioning experiments on TUM RGB-D dataset, which is the mostly used SLAM benchmark in literature. It provides a variety of data sequences with precise ground-truth trajectories. In this experiment, we selected Handheld SLAM, Robot and Structure vs. Texture categories including 11, 1 and 8 subsequences respectively. We test ORB-SLAM2, DXSLAM and TRVO on the same device, and compare the positioning accuracy of these three systems in different sequences. **Fig. 6** shows the motion trajectory and relative pose error drawn by evo tool. Due to space limit, we only show the trajectory on fr1_desk and fr2_pioneer_360 sequence. It can be seen that ORB-SLAM2 and DXSLAM tracked lost for a short time on fr2_pioneer_360 sequence, because there are not enough valid keypoints. While TRVO generated a complete trajectory thanks to LoFTR can provide dense matches in challenging environment. It should be emphasized that TRVO is only a simple visual odomery, its absolute positioning error is not as good as the other two algorithms due to error accumulation. However, TRVO is more robust and has higher relative positioning accuracy, which can be confirmed in **Table. 1**. **Table. 1** lists the performance of the three systems on all test sequences. Note that we use relative pose error to evaluate the positioning accuracy. Because TRVO is just a visual odometry, which equivalent to the front end of the visual SLAM system, there will be a certain cumulative error in pose estimation, while the modules of back-end optimization, loop detection and re-localization in ORB-SLAM2 and DXSLAM will reduce the cumulative error to a certain extent and make the absolute pose error lower. Therefore, in order to make the comparison as fair as possible, we choose the relative pose error as the evaluation criteria. From the table, we can see that TRVO has the lowest relative pose error in most of sequences.

4.3. Localization on corridor environment

We collected a sequence of images collected by Intel RealSense D435i camera in corridor with texture-less walls, corresponding to the texture-less scene in **Fig. 5**. We held the camera and walk around the corridor, we show the estimated trajectory in **Fig. 7**. ORB-SLAM2 and DXSLAM failed to generate a complete trajectory, they tracked lost soon after initialize. While TRVO completed the whole process as we expected.

5. CONCLUSIONS

In this paper, we propose TRVO, a visual odometry based on learning method. The TRVO get feature matches between images form LoFTR network, and uses nonlinear optimization method to minimize the reprojection error to estimation the pose of the camera. This learning-based feature matching network incorporates transformer structure, which can make full use of the global information of the image. Furthermore, our experiments show that TRVO system gives pose estimates with much lower relative pose error and it is more robust in challenging environments with low-texture or repetitive patterns, while hand-crafted based method often fails in such a scenario. However, we noted that the LoFTR network is time-consuming compared with hand-crafted methods, and the real-time performance cannot be guaranteed. In addition, LoFTR is a detector-free matching approach, which directly regresses matches from a pair of images. As such, the feature matching results generated by image pairs (A, B) and (B, C) may be slightly different for the same frame image B. As a result, TRVO cannot do back-end optimization, loop detection and relocation as ORB-SLAM and DXSLAM, it is the common inferiority of all detector-free methods in visual odometry. We leave it as our future work to accelerate the inference time and add descriptors for the network.

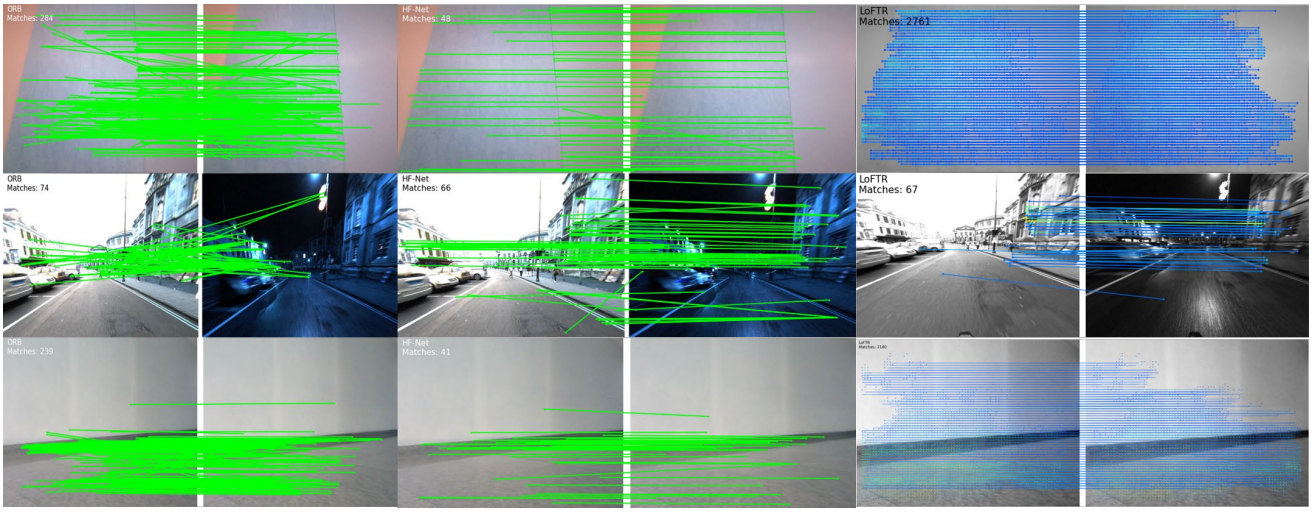
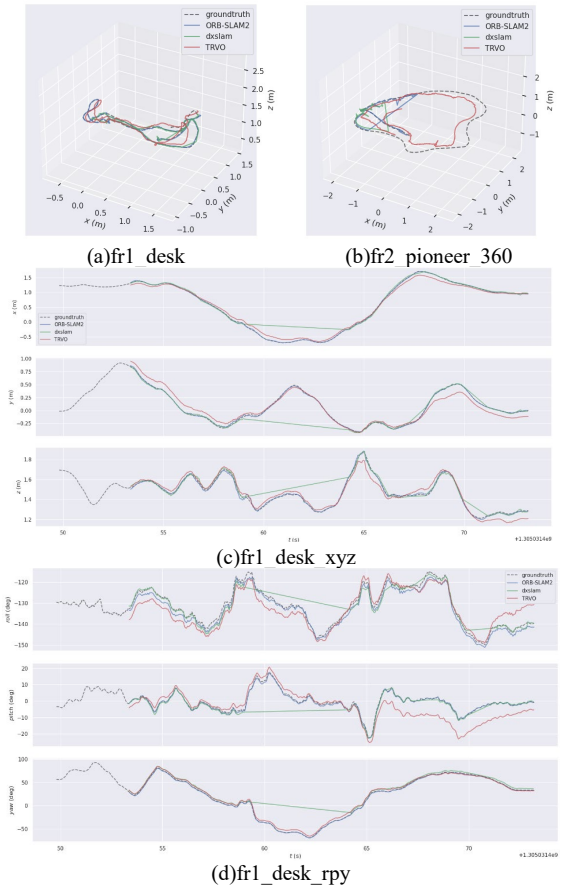


Fig. 5 Feature matching results. We show three queries and the retrieved images for ORB feature (left), HF-Net (middle) and LoFTR network(right).



(a)fr1_desk (b)fr2_pioneer_360

(c)fr1_desk_xyz

(d)fr1_desk_rpy

Fig. 6 Estimated trajectory on TUM dataset

Table. 1 Mean relative pose error calculated by evo.

dataset	evo_rpe (trans_part and rot_part, unit-less)		
	ORB-SLAM2	**DXSLAM**	**TRVO**
fr1_360	0.240654	0.013515	**0.012753**
fr1_desk	0.014376	0.016301	**0.012022**
fr1_desk2	0.015431	0.016248	**0.012817**
fr1_floor	0.006921	**0.005871**	0.007212
fr1_room	0.013139	0.011891	**0.010296**
fr2_360_h	0.017897	**0.013530**	0.021250
fr2_360_k	0.014510	**0.013143**	0.016878
fr2_desk	**0.006578**	0.007624	0.006971
fr2_l_n_l	**0.017060**	0.018296	0.026593
fr2_l_w_l	0.020013	**0.015282**	0.022142
fr3_l_o_h	0.007284	0.007479	**0.007094**
fr2_p_360	0.023103	0.026169	**0.022892**
fr3_n_n_f	0.028068	-[a]	**0.023827**
fr3_n_n_n_w	-	-	**0.027900**
fr3_n_t_f	0.030258	0.028255	**0.022806**
fr3_n_t_n_w	0.014841	0.014618	**0.014105**
fr3_s_n_f	0.010663	-	**0.007317**
fr3_s_n_n	0.019516	-	**0.009913**
fr3_s_t_f	0.012829	0.012706	**0.010329**
fr3_s_t_n	0.012155	0.011895	**0.009827**

a. '-' means the algorithm fail to track all the frames

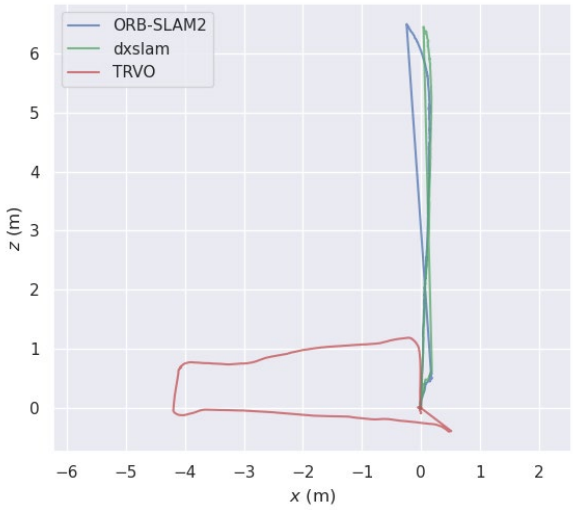


Fig. 7 Estimated trajectory on corridor

Acknowledgements

The project is supported by the National Science Foundation of China (Grant No. 41874034), the National key research and development program of China (Grant No.2016YFB0502102), the Beijing Natural Science

Foundation (Grant No. 4202041), and the Aeronautical Science Foundation of China.

REFERENCES:

[1] Dai Z , X Huang, Chen W , et al. A Comparison of CNN-Based and Hand-Crafted Keypoint Descriptors[C]// 2019 International Conference on Robotics and Automation (ICRA). 2019.

[2] Lowe D G . Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.

[3] Bay H , Ess A , Tuytelaars T , et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision & Image Understanding, 2008, 110(3):346-359.

[4] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, 2011, pp. 2564-2571.

[5] Sun J , Shen Z , Wang Y , et al. LoFTR: Detector-Free Local Feature Matching with Transformers[J]. 2021.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. NeurIPS, 2017.

[7] Davison, Andrew, J, et al. MonoSLAM: Real-Time Single Camera SLAM.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007.

[8] Kameda Y . Parallel Tracking and Mapping for Small AR Workspaces (PTAM) Augmented Reality[J]. Journal of the Institute of Television Engineers of Japan, 2012, 66(1):45-51.

[9] Mur-Artal R , Montiel J M M , Tardos J D . ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.

[11] Campos C , Elvira R , JJG Rodríguez, et al. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM[J]. 2020.

[12] Tong, Qin, Peiliang, et al. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator[J]. IEEE Transactions on Robotics, 2018.

[13] Tang J , Ericson L , Folkesson J , et al. GCNv2: Efficient Correspondence Prediction for Real-Time SLAM[J]. IEEE Robotics and Automation Letters, 2019.

[14] Li D , Shi X , Long Q , et al. DXSLAM: A Robust and Efficient Visual SLAM System with Deep Features[J]. 2020.

[15] Sarlin P E , Cadena C , Siegwart R , et al. From Coarse to Fine: Robust Hierarchical Localization at Large Scale[J]. 2018.

[16] Detone D , Malisiewicz T , Rabinovich A . SuperPoint: Self-Supervised Interest Point Detection and Description[J]. 2017.

[17] D Detone, Malisiewicz T , Rabinovich A . Toward Geometric Deep SLAM. 2017.

[18] Sarlin P E , D Detone, Malisiewicz T , et al. SuperGlue: Learning Feature Matching with Graph Neural Networks[J]. arXiv, 2019.

[19] Zhou Q , Sattler T , Leal-Tai Xe L . Patch2Pix: Epipolar-Guided Pixel-Level Correspondences[J]. 2020.

[20] Fischler M A , Bolles R C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography - ScienceDirect[J]. Readings in Computer Vision, 1987:726-740.

[21] Kummerle R , Grisetti G , Strasdat H , et al. G2o: A general framework for graph optimization[C]// IEEE International Conference on Robotics & Automation. IEEE, 2011.

[22] Sattler T , Maddern W , Toft C , et al. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.