3OFRR-SLAM: Visual SLAM with 3D-Assisting Optical Flow and Refined-RANSAC

Yujia Zhai, Fulin Tang and Yihong Wu

November 10, 2021

3OFRR-SLAM: Visual SLAM with 3D-assisting Optical Flow and Refined-RANSAC*

Yujia Zhai[1,2][0000-0001-9729-484X], Fulin Tang[1][0000-0002-8474-2671], and Yihong Wu[1][0000-0002-9595-7686]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China. fulin.tang, yhwu@nlpr.ia.ac.cn.
[2] Institute of Semiconductors, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, China.

Abstract. To perform navigation or AR/VR applications on mobile devices, SLAM is expected to be with low computational complexity. But using feature descriptors restricts the minimization and lightweight of a SLAM system. In this paper, we propose a lightweight monocular SLAM system called 3OFRR-SLAM, which is precise, fast, and achieves real-time performance on CPU and mobile phones. It integrates a 3D-assisting optical flow tracker, uses a local map to provide prior information for optical flow, and improves the Lucas-Kanade algorithm, which makes data association fast and reliable. To further eliminate outliers of data association, we propose a novel Refined-RANSAC, improving the accuracy of camera pose estimation without taking much extra time cost. We evaluate our system on TUM-RGBD dataset and real-world data. The results demonstrate that our system obtains an outstanding improvement in both speed and accuracy compared with current state-of-the-art methods ORB-SLAM2 and DSO. Moreover, we transplant our system to an android-based smartphone and show the application for augmented reality (AR).

Keywords: Visual Localization · Fast Tracking · SLAM.

1 Introduction

Over the past decades, simultaneous localization and mapping(SLAM) [3] has made rapid progress. With visual SLAM technology getting mature, it has been applied in robotics, unmanned driving and AR/VR widely [23]. Several sensor types can be used as input of visual SLAM, such as monocular cameras [12] [6], stereo cameras [22] and RGB-D cameras [9]. In consideration of the low cost and easy deployment of the monocular camera, we focus on monocular visual SLAM in this paper.

For visual SLAM, one of the key problems is data association, which has a decisive influence on the efficiency and accuracy of the visual localization

* Supported by National Natural Science Foundation of China under Grant Nos.62002359, 61836015 and the Beijing Advanced Discipline Fund under Grant No.115200S001.

and reconstruction [26]. Based on the geometric features of image points, data association methods can be divided into two types. One is to construct data association by calculating specific descriptors of feature points. Although high accuracy and robustness can be obtained, even fast feature descriptors such as ORB [20] may decrease the real-time performance of SLAM systems in the case of high frame rates and high image resolution. The other is to detect corners and utilize sparse optical flow to construct data association. Compared with the first one, it is faster but more likely to cause incorrect feature correspondences. Another mainstream method of data association is the direct method, which directly minimalizes the photometric error by solving a nonlinear optimation instead of relying on geometric features. As it avoids the complex calculation of descriptor matching, it obtains high efficiency. However, the optimization of sliding windows still requires a lot of computing power, which also limits the application of this method on the mobile terminals.

To address these problems, we propose 3OFRR-SLAM, a lightweight and accurate monocular SLAM system, which combines a 3D-assisting optical flow tracker that can give one-to-one correspondences for accurate and fast data association. And to further reject outliers of 3D-2D correspondences obtained from the proposed tracker, we propose a novel Refined-RANSAC method to refine the estimation of camera poses, where we also give a criteria function to choose 3D information with high quality.

The main contributions of this paper are as follows:

- A lightweight monocular visual SLAM system integrates a 3D-assisting optical flow tracker.
- A novel Refined-RANSAC method is proposed to better eliminate outliers. Compared with standard-RANSAC, it makes camera pose estimation more accurate without taking much extra time cost.
- We demonstrate on TUM-RGBD [**?**] dataset that our system outperforms the state-of-art systems in accuracy and speed, and implement our system on the mobile terminal for visual localization and AR applications.

The rest of this paper is organized as follows. In section 2 we discuss the related work. The framework of the proposed system is shown in Section 3. Section 4 provides the details of the tracking thread and the mapping thread is described in Section 5. Section 6 provides qualitative and quantitative experimental results. And the conclusions and future work are given in Section 7.

2 Related Work

At present, the mainstream visual SLAM methods can be roughly divided into three categories: filter-based visual SLAM, keyframe-based visual SLAM, and direct-based visual SLAM.

Filter-based visual SLAM uses a Gaussian probability model to express the system state at each moment and continuously update it. Davison first proposed MonoSLAM [2], a real-time SLAM system using a monocular camera in 2003.

MonoSLAM is implemented using the Extended Kalman Filter (EKF) [25] under a probability framework. The computational complexity of MonoSLAM is very high, which makes it difficult to apply on a large scale. Paz et al [17] proposed a divide-and-conquer EKF-SLAM method to reduce the amount of calculation. Filter-based visual SLAM has more entangled data association and thus is more easy to drift.

It is proved in [21] that the keyframe-based visual SLAM outperforms filter-based visual SLAM. Klein et al [10] proposed and open-sourced the first keyframe-based monocular visual SLAM system called PTAM in 2007, and transplanted it to the iPhone 3G in 2009 [11]. A classic two-thread framework is proposed in PTAM, which performs tracking and mapping as two independent tasks in two parallel threads. ORB-SLAM [14] proposed by Mur-Artal et al in 2015 gets improvement from the FAST features [19] and the ORB [20] descriptors as while as the addition of the loop detection module. They further proposed ORBSLAM2 [15] in 2017, which is an extension from ORB-SLAM.

Visual direct SLAM directly optimizes the photometric error instead of considering the geometric information. It attains better robustness in the case of weak texture and images blurred. DTAM [16] is a dense visual SLAM system based on the direct method proposed in 2011. It constructs a dense depth map of keyframes by minimizing the energy function of the global space specification. DTAM is computationally intensive and requires GPU parallel computing. LSD-SLAM [5] restores the depth values of some pixels in the image to get a semi-dense model, which can run in real time on the CPU and smartphones. DSO [4] is a sparse direct visual odometry, which combines the photometric calibration model to improve the robustness. DSO can also run in real time on the CPU. What's more, SVO [7] and SVO2 [8] are a kind of semi-direct visual odometry and use a combination of feature points and direct methods. In the tracking thread, they extract FAST [19] corners and track them using the direct method while using a depth filter [18] to recover the depths. Since they avoid to calculate a large number of descriptors, they can be extremely fast.

These methods either have entangled data association or use image descriptors to associate data. To make SLAM more lightweight along with better accuracy, we design a visual odometry based on the optical flow, utilizing the information of the local map and the forward poses to optimize and accelerate the optical flow tracking. Besides, we solve the camera pose by the proposed Refined-RANSAC, which is a promotion of Lebeda's LO-RANSAC [**?**]. But instead of processing the data completely based on randomness without bias, we introduce the information of the local map to give the data points an estimated weight in voting, which makes camera pose estimation more accurate without taking much extra time cost.

3 SYSTEM OVERVIEW

The system overview of 3OFRR-SLAM is shown in Fig.1. Tracking and reconstructing run in two separate threads. The tracking thread implements the pro-

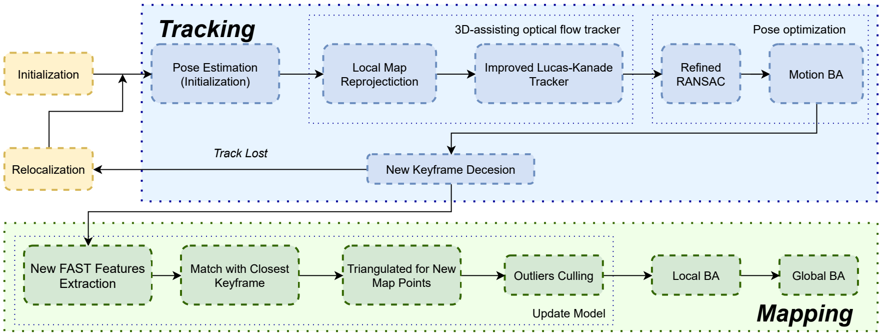

Tracking
3D-assisting optical flow tracker
Pose optimization
Initialization
Pose Estimation (Initialization)
Local Map Reprojectiction
Improved Lucas-Kanade Tracker
Refined RANSAC
Motion BA
Relocalization
Track Lost
New Keyframe Decesion
New FAST Features Extraction
Match with Closest Keyframe
Triangulated for New Map Points
Outliers Culling
Update Model
Local BA
Global BA
Mapping

Fig. 1. The framework of 3OFRR-SLAM

posed 3D-assisting optical flow tracker to get an accurate estimation of the camera pose for each frame. First we detect FAST corners on the current frame $\mathbf{K_j}$ and get an initial pose prediction from the last frame $\mathbf{K_{j-1}}$. Then we use an improved LK optical flow tracker to reproject the local map into the current frame $\mathbf{K_j}$ with the initial pose to obtain 3D-2D correspondences, and estimate the camera pose with the proposed Refined-RANSAC. The pose will be further optimized by a local BA(bundle adjustment). If the current frame is determined to be a keyframe, we will add it to the keyframe sequence, in which it will be waited to be inserted to the mapping thread.

The mapping thread incrementally reconstructs the 3D structure of the surroundings. Map points will be produced by triangulation with the 2D-2D correspondences found between the current keyframe and its nearst keyframe. Afterwards, a local BA is performed to refine the new reconstructed points and then a global BA will be performed within several selected representative keyframes.

4 Tracking

4.1 3D-assisting Optical Flow Tracker

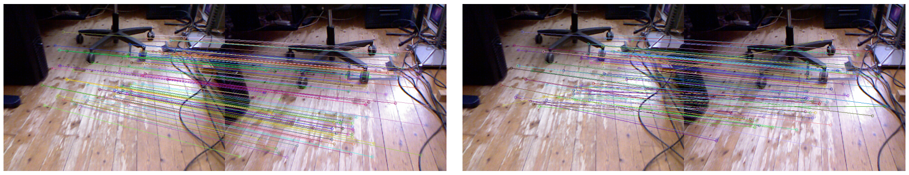
Those 3D points which may be visible will be chosen to be tracked by the proposed 3D-assisting optical flow tracker to get 3D-2D correspondences.

Considering that the disparity between the reference keyframe and the current frame may be relatively large, the source image patch around $\boldsymbol{m_{ij}}$ and the target image patch around $\boldsymbol{m_{ic}}$ may be quite different and remote. If we directly calculate the optical flow, it could hardly find the right correspondences. Firstly we use LK optical flow to track points between the last frame and the current frame to get an intial pose prediction of the current frame $\mathbf{K_j}$. According to 3D-2D correspondences in the current frame, the initial pose of $\mathbf{K_j}$ can be estimated by solving the PnP problem. Afterwards an affine transformation is performed to correct the source image patch. Then we set the initial search position for the optical flow tracker by projecting the selected 3D points onto the current frame with the predicted initial camera pose. This strategy is a one-to-one correspondence way and thus makes the initial search position closer to the real value, which accelerates the iteration and improves the matching accuracy. That's why we call it the 3D-assisting optical flow tracker. It can be seen in Fig.2 that the mismatchs of our method are much less than the ordinary LK optical flow.

Starting from the given initial positions, the 3D-assisting optical flow tracker iteratively searches the correspondence of the source patch in the target image by minimizing the sum of squared difference (SSD) between them. For each iteration, the following photometric residual is minimized:

$$\begin{aligned}
&\sum_{x=-r}^{r} \sum_{y=-r}^{r} \left(I(x,y) - J(x + x_0 + dx, y + y_0 + dy)\right)^2 \\
&= \sum_{p=0}^{(2r+1)^2} \left(I(p) - J(p)|_{(x_0+dx, y_0+dy)}\right)^2 ,
\end{aligned} \tag{1}$$

where r is the radius of the image patch, I and J are the brightness of each pixel in the source patch and the target patch, (x_0, y_0) is the position after the last iteration and the start position of this iteration and (dx, dy) is the required offset.



(a) The proposed 3D-assisting optical flow tracker

(b) The ordinary LK optical flow

Fig. 2. Performances of optical flow tracker on TUM- RGBD dataset.

To mitigate the influence of lighting variation while speeding up the convergence, unlike the LK optical flow method that modifies the residual representation and introduces new parameters, we adopt a direct method: normalize the source image patch to get $I^{'}(p)$, and the image patch at the beginning of this iteration in the target image is also normalized to get $J^{'}(p)$, then the solution of (dx, dy) can be expressed as:

$$\begin{aligned}
\begin{bmatrix} dx \\ dy \end{bmatrix} &= \begin{bmatrix} \sum_p I_x^{'}(p)^2 & \sum_p I_x^{'}(p) I_y^{'}(p) \\ \sum_p I_x^{'}(p) I_y^{'}(p) & \sum_p I_y^{'}(p)^2 \end{bmatrix}^{-1} \\
&\quad \cdot \begin{bmatrix} \sum_p I_x^{'}(p) \left(I(p)^{'} - J^{'}(p)|_{(x_0,y_0)}\right) \\ \sum_p I_y^{'}(p) \left(I(p)^{'} - J^{'}(p)|_{(x_0,y_0)}\right) \end{bmatrix} ,
\end{aligned} \tag{2}$$

where$(I_x^{'}, I_y^{'})$ are the normalized brightness of the source patch.

To further adapt to the situation where the position of the target image patch is greatly different from the predicted patch, we build a 4-level image pyramid and obtain the target position from coarse to fine.

4.2 Refined-RANSAC and Camera Pose Estimation

In order to further improve the accuracy of the pose estimation, we study a novel Refined-RANSAC to remove 2D-3D correspondence outliers.

The standard RANSAC algorithm randomly selects several data elements from the input data set, then repeatedly solves a model that fits the chosen samples. It considers that each data point has an equal confidence when evaluating the estimated model. Therefore, the number of the data elements that the model can fit within a given tolerance is used as the evaluation criteria of model quality. And the model with the most inliers will be returned. This leads to the fact that the performance of standard RANSAC method will decrease rapidly as the outliers ratio increases. And it is very sensitive to the threshold boundaries for dividing inliers and outliers. Meanwhile, the standard RANSAC is based on the assumption that all-inlier samples lead to the optimal solution. However, this assumption has been observed to be not valid in practice as pointed out in [1] [24].

To this end, we propose the Refined-RANSAC to further reject outliers in pose estimation. Compared to the standard RANSAC, the proposed Refined-RANSAC includes two new processes.

1. **Local Optimization**: after a potential model is found by the standard RANSAC, we run an additional local optimization step on it.

2. **Weighted Voting**: data points are assigned with different voting weights in the model evaluation according to their reliability.

We use the past recurrence rate of a map point ω as an index to evaluate the reliability of the map point. The recurrence rate ω is given by:

$$\omega = N^{'}/N, \tag{3}$$

where N is the number of past frames at which the point can be observed and $N^{'}$ is the number of the times that the map point and its corressponding 2D point stay within inliers after local bundle adjustment. And the score $\mathcal{E}_M$ of a model can be calculated as follows:

$$\mathcal{E}_M = \sum_{i=1} max(\omega \cdot |p_i^m - p_i|, Thr_{error}), \tag{4}$$

where p_i is the real position of an image point, p_i^m represents the reprojected position of p_i using model M, and Thr_{error} is a given threshold to limit the impact of a single data point.

The whole process of the proposed Refined-RANSAC is summarized in **Algorithm** 1 and the local optimization step is summarized in **Algorithm** 2, where LSq is short for least squares solution, $\mathcal{L}$ is the set of inliers of 3D-2D correspondence and M represents the estimated model of camera pose (the best found, the best

Algorithm 1 Refined-RANSAC

```
Require: $\mathcal{I}, \omega, \theta$
$\mathcal{I} \leftarrow$ input samples
$\omega \leftarrow$ the weights of input data
$\theta \leftarrow$ the inlier-outlier error threshold
for $k = 1 \rightarrow \mathrm{K}(\mathcal{I})$ do
    $\mathcal{S}_k \leftarrow$ randomly drawn minimal sample from $\mathcal{I}$
    $M_k \leftarrow$ model estimated from sample $\mathcal{S}_k$
    $\mathcal{E}_k \leftarrow$score_model$(M_k, \omega, \theta)$
    if $\mathcal{E}_k > \mathcal{E}_s^*$ then
        $M_s^* \leftarrow M_k; \mathcal{E}_s^* \leftarrow \mathcal{E}_k$
        $M_{LO}, \mathcal{E}_{LO} \leftarrow$ run Local Optimization $(M_s^*, \omega, \theta)$
        if $\mathcal{E}_{LO} > \mathcal{E}^*$ then
            $M^* \leftarrow M_{LO}; \mathcal{E}^* \leftarrow \mathcal{E}_{LO}$
            update K
        end if
    end if
end for
return $M^*$
```

from local optimization). The find_inliers function evaluates the samples with input model M, and returns the subset of inliers whose errors are smaller than threshold θ.

As shown in **Algorithm** 1, firstly we execute an outer iteration with a minimal solver to find a potential best model. The minimal solver is used because the input dataset here has not been filtered and may have a relatively larger outlier ratio. Then we run the local optimization phase. The samples input here are only chosen from $\mathcal{E}_s$ (the inliers to the model found by a minimal solver under a slightly bigger in-out-threshold). As the sampling is running on so-far-inner data, in this iteration a non-minimal solver can be performed to introduce more information, such as the nonlinear optimization we used.

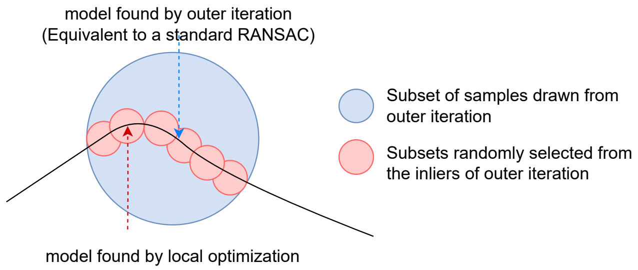

model found by outer iteration
(Equivalent to a standard RANSAC)
Subset of samples drawn from outer iteration
Subsets randomly selected from the inliers of outer iteration
model found by local optimization

Fig. 3. Schematic of the Local Optimization

Substantially, the inner local optimization step aims to refine the estimated model by solving the same ploblem again within a smaller area which is verified to be reliable by the outer interation. Fig.3 is a schematic of the process. The

Refined-RANSAC algorithm is very stable and is insensitive to the choice of the inlier-outlier threshold. It offers a significantly better initial value point for bundle adjustment. The quantitative evaluation of the performance of Refined-RANSAC can be seen in Section 6, which can be seen in Table 1.

The pose estimation obtained after the Refined-RANSAC solver will be refined further by a non-linear optimizer with 3D-2D correspondences.

Algorithm 2 Local Optimization

```
Require: $M_s, \omega, \theta, , m_\theta$
$M_s \leftarrow$ model estimated by outer iteration
$\omega \leftarrow$ the weights of input data
$\theta \leftarrow$ the inlier-outlier error threshold
$m_\theta \leftarrow$ the threshold multiplier
$\mathcal{I}_s \leftarrow$ find_inliers $(M_s, \theta)$
for $i = 1 \rightarrow iters1$ do
    $\mathcal{S}_{is} \leftarrow$ sample of size $s_{is}$ randomly drawn from $\mathcal{I}_s$
    $M_{is} \leftarrow$model estimated from $\mathcal{S}_{is}$ by least squares solution
    $\boldsymbol{\theta}' \leftarrow m_\theta \cdot \boldsymbol{\theta}$
    $\mathcal{I}' \leftarrow$ find_inliers $(M_{is}, \theta')$
    $M' \leftarrow$ model estimated by nonlinear optimization on $\mathcal{I}'$
    $\mathcal{E}' \leftarrow$ score_model $(M', \omega, \theta')$
    if $\mathcal{E}' > \mathcal{E}'^*$ then
        $M_s^* \leftarrow M'$
    end if
    $M_r \leftarrow$ the best of $M^{'}$
end for
return the best of $M_r$, with its inliers
```

4.3 Relocalization

If the tracking quality is poor for consecutive 5 frames, tracking is assumed to be lost and the relocalization will be performed in the next frame. The relocalization process firstly searches for the 3D-2D correspondences between the global map and current frame through a random forest. Then the camera pose is calculated by EPnP algorithm and the current camera pose is retrieved in the global map.

5 Mapping

The addition of new keyframes brings new information to update the map. We detect FAST feature points on the newly added keyframe and select those further from the observation of existing map points as new features, then search for their correspondences on the nearest keyframes. Matching search is performed along the epipolar line with a cross-check matching method: two feature points are considered to be a valid pair only if they are both the most similar feature points of each other. Once the correspondences are obtained, the new map points can be triangulated.

After updating the model, we perform a local bundle adjustment [13] to optimize the newly reconstructed map. And then a global bundle adjustment is applied. Considering the efficiency of the system in a long image sequence, we select some representative keyframes through the covisible relationship and optimize the map points and the poses of the selected keyframes together.

6 EXPERIMENT

We perform experiments on a public dataset and in a real world environment to evaluate the proposed 3OFRR-SLAM system. We carry out all experiments with an Intel Core i7-8750H CPU (12 cores@ 2.20GHz) and 32 GB RAM. Our system is compared with two state-of-the-art methods: ORB-SLAM2 and DSO. Additionally, we port the proposed system to two applications, including an implementation for augmented reality and an app running on an android mobile device.

Table 1. Localization error of each frame comparison in the TUM RGB-D dataset

Sequence	3OFRR-SLAM				ORB-SLAM2		DSO	
	Standard RANSAC		Local RANSAC					
	ATE (m)	RPE (deg/m)	ATE (m)	RPE (deg/m)	ATE (m)	RPE (deg/m)	ATE (m)	RPE (deg/m)
fr1/xyz	0.0131	0.4912	**0.0083**	**0.4822**	0.0102	0.5256	0.0202	0.9982
fr2/xyz	0.0312	0.7088	0.0318	0.7275	**0.0251**	**0.0242**	0.0416	0.554
fr2/rpy	0.0092	1.3305	**0.0068**	**1.3178**	CNI[1]	CNI	0.0152	2.9585
fr1/desk	0.0201	1.3744	**0.0169**	1.2780	0.0190	**1.2556**	0.0285	1.8569
fr2/desk	0.0362	0.6989	0.0220	0.6875	**0.0116**	**0.2443**	0.0224	0.6899
fr3/long_office	0.0322	0.2366	**0.0275**	**0.2091**	0.0434	0.2334	0.0855	0.7341
fr3/sitting_halfsphere	0.0551	0.9002	0.0379	0.8452	**0.0142**	**0.4266**	0.0410	0.9233
fr3/sitting_xyz	0.0180	0.3170	**0.0176**	**0.3166**	0.0268	0.3646	0.0232	1.3140
fr3/walking_halfsphere	0.1722	2.1220	**0.0990**	**1.8201**	0.1606	3.1870	0.1953	3.2845
fr2/desk_with_person	0.0090	0.3109	**0.0064**	**0.2815**	0.0072	0.2955	0.0288	1.0020
fr3/str_tex_near	0.0575	0.5090	**0.0175**	**0.4732**	0.0188	0.5323	0.0219	1.2091
fr3/str_tex_far	0.0309	0.5466	0.0184	0.5533	**0.0091**	0.2108	0.1056	2.6555
fr3/nostr_tex_near_withloop	0.0303	0.3810	**0.0247**	**0.3099**	0.0262	0.3217	0.0426	0.6713
fr3/nostr_tex_far	0.1022	3.2988	**0.0690**	**2.4750**	AD[2]	AD	0.8522	2.9666

[1] CNI: cannot initialize

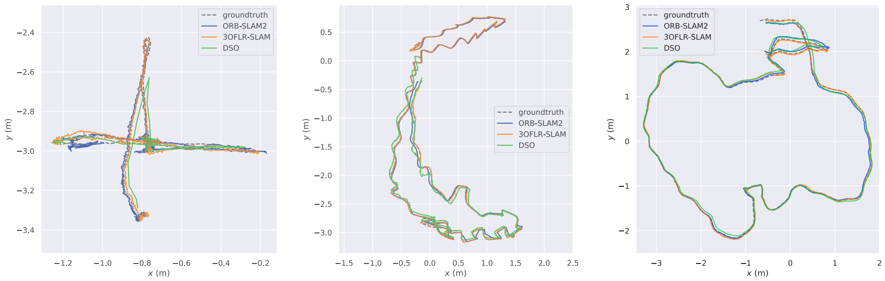
[2] AD:ambiguity detected

6.1 Evaluation on TUM-RGBD Dataset

We evaluate our system on the public TUM-RGBD dataset, which contains indoor sequences from RGBD sensors grouped in several categories and provides the ground truth of the camera pose for each frame. It is widely used to evaluate the SLAM or odometry systems. Since 3OFRR-SLAM is based on a monocular camera, we only use the RGB images as input.

ORB-SLAM2 and DSO are chosen to be the state-of-the-art systems of the feature-based methods and the direct methods respectively. The experimental results of them are obtained by running the open-source codes. To be fair, the ORB-SLAM2 is loop closure-disabled. We run each sequence for 5 times and pick the median result. The camera poses of all frames are recorded.

We adopt absolute trajectory error (ATE) and relative pose error (RPE) to conduct the quantitative evaluation of the system. The ATE directly calculates the difference between the ground-truth and the estimated camera poses,which reflects the overall performance of system. The relative pose error describes the pose difference of two frames during a fixed time interval. It reflects the drift of the system. The Root-Mean-Square Error (RMSE) of ATE and RPE is shown in TABLE 1, which is calculated by the benchmark tool [27]. It can be seen in TABLE 1 that our system outperforms ORB-SLAM2 and DSO on most sequences. That benefits from the accurate position from the proposed optical flow tracker, and the subsequent Refined-RANSAC gives a reliable filtering of data associations. The severe jitter and rapid moving on some sequences (such as fr2/xyz) will cause motion blur, and make most of the motion assumptions invalid, which makes optical flow tracking and the direct method performs terrible. But it has relatively little impact on the ORB-SLAM2, which adopts robust ORB feature to get correspondences.



(a) fr3/sitting_xyz (b) fr2/desk_with_person (c) fr3/long_office

Fig. 4. Estimated trajectories by 3OFRR-SLAM, ORB-SLAM2 and DSO on TUM-RGBD dataset.

Fig.4 shows the estimated trajectories by our method, ORB-SLAM2 and DSO on three different TUM-RGBD sequences. It can be seen clearly that our estimated trajectories are smoother than those of ORB-SLAM2 and DSO with fewer sudden jumps.

Running time is an important factor of the performance of the online system. Fig.5 shows the time cost comparisons of our camera pose tracking method with ORB-SLAM2 and DSO on the fr3/long_office sequence. As the proposed method is based on the optical flow, it preserves from the calculation and matching of

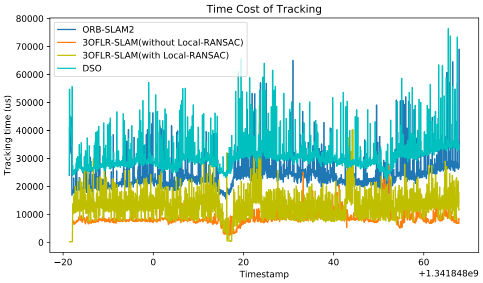


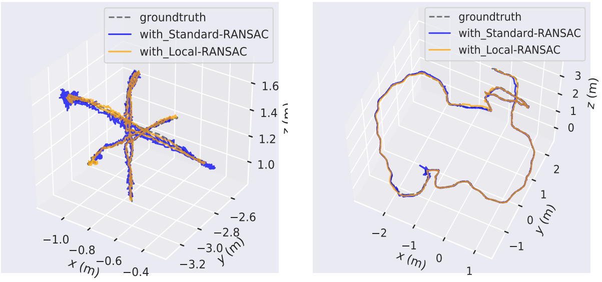
Fig. 5. Tracking timecost of each frame on the sequence fr3/long_office of TUM-RGBD dataset

descriptors adopted in ORB-SLAM2 and the iteratively optimization of the sliding keyframe window adopted in DSO. The reduced computational complexity brings a noticeable speed increase. It can be seen that our speed can generally reach 2-3 times of ORB-SLAM2 and DSO.

6.2 Evaluation of Refined-RANSAC

The proposed Refined-RANSAC brings obvious improvement in terms of accuracy compared to standard RANSAC,which can be seen in Table 1. And Fig.6 shows that the pose trajectory with Refined-RANSAC is more smooth and closer to the groundtruth.

The experiment on TUM-RGBD dataset demonstrates that Refined-RANSAC does not need much extra time cost in contrast to the standard RANSAC. The reason is that more correct inliers can trigger the stopping criterion earlier and the improvement of the initial value can accelerate the convergence of the subsequent optimization. It can be seen in Fig.5 that the addition of the Refined-RANSAC method does not affect the speed of camera pose tracking, and even speeds up the convergence of pose optimization.



(a) fr3/sitting_xyz (b) fr3/long_office

Fig. 6. Performances of the proposed Refined-RANSAC compared with standard-RANSAC on TUM- RGBD dataset.

6.3 Application

1) *AR with a Hand-held Camera*

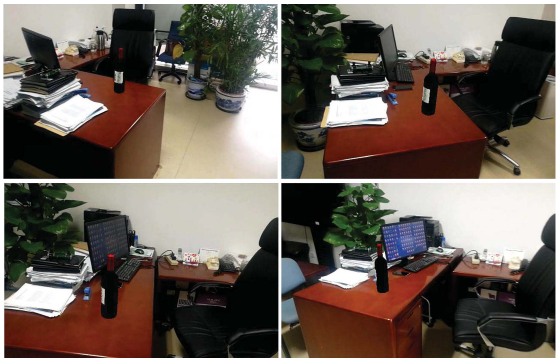


Fig. 7. An implementation for augmented reality: a virtual 3D model of a wine bottle is projected on the desk in the real office scene. Although the camera moves, the bottle remains still stable.

We present a simple AR application with a hand-held monocular camera to exhibit the accuracy and robustness of our system. The performance can be seen in Fig.7. We add a virtual 3D model of a wine bottle to the real scene in the office and place it on the desk. It can be seen that the red wine bottle on the desk remains stable as the hand-held camera rotates and moves, demonstrating the high visual localization accuracy of our SLAM system.

2) *Implementation on Mobile Device*

As 3OFRR-SLAM is lightweight, we transplant it to mobile devices and test its performance in real-world indoor scene. It can run in real time on a Huawei P9 smartphone, using images with 30 Hz and 640 × 480 resolution. Fig.8 shows the performance of the app.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a lightweight SLAM system that uses optical-flow to solve data association instead of the common-used descriptor-based method or direct method. The system is composed of an improved 3D-assisting optical flow tracker and a novel Refined-RANSAC algorithm that combines the information of the local map to further eliminate the outliers and improve the camera pose estimation. Experiments show that the proposed SLAM system has superior performances in terms of accuracy and speed than state-of-the-art methods. And it is proved that the system we proposed can run in real time on a small mobile terminal. In the future, we will add closure loop detection to deal with large city environments and fuse IMU information to assist visual localization.

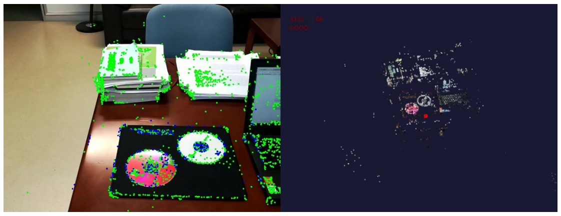


Fig. 8. APP of 3OFRR-SLAM running on a mobile device in a real-world indoor scene, with 30 Hz and 640 × 480 resolution images as input.

References

1. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Joint Pattern Recognition Symposium. pp. 236–243. Springer (2003)
2. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE transactions on pattern analysis and machine intelligence **29**(6), 1052–1067 (2007)
3. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. IEEE robotics & automation magazine **13**(2), 99–110 (2006)
4. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **40**(3), 611–625 (2017)
5. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014)
6. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: Proceedings of the IEEE international conference on computer vision. pp. 1449–1456 (2013)
7. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 15–22. IEEE (2014)
8. Forster, C., Zhang, Z., Gassner, M., Werlberger, M., Scaramuzza, D.: Svo: Semidirect visual odometry for monocular and multicamera systems. IEEE Transactions on Robotics **33**(2), 249–265 (2016)
9. Kerl, C., Stuckler, J., Cremers, D.: Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In: Proceedings of the IEEE international conference on computer vision. pp. 2264–2272 (2015)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. pp. 225–234. IEEE (2007)
11. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: 2009 8th IEEE International Symposium on Mixed and Augmented Reality. pp. 83–86. IEEE (2009)
12. Li, X., Ling, H.: Hybrid camera pose estimation with online partitioning for slam. IEEE Robotics and Automation Letters **5**(2), 1453–1460 (2020)
13. Lourakis, M.I., Argyros, A.A.: Sba: A software package for generic sparse bundle adjustment. ACM Transactions on Mathematical Software (TOMS) **36**(1), 1–30 (2009)
14. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)

15. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
16. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: 2011 international conference on computer vision. pp. 2320–2327. IEEE (2011)
17. Paz, L.M., Jensfelt, P., Tardos, J.D., Neira, J.: Ekf slam updates in o (n) with divide and conquer slam. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. pp. 1657–1663. IEEE (2007)
18. Pizzoli, M., Forster, C., Scaramuzza, D.: Remode: Probabilistic, monocular dense reconstruction in real time. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 2609–2616. IEEE (2014)
19. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European conference on computer vision. pp. 430–443. Springer (2006)
20. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)
21. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual slam: why filter? Image and Vision Computing **30**(2), 65–77 (2012)
22. Tang, F., Li, H., Wu, Y.: Fmd stereo slam: Fusing mvg and direct formulation towards accurate and fast stereo slam. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 133–139. IEEE (2019)
23. Tang, F., Wu, Y., Hou, X., Ling, H.: 3d mapping and 6d pose computation for real time augmented reality on cylindrical objects. IEEE Transactions on Circuits and Systems for Video Technology **30**(9), 2887–2899 (2019)
24. Tordoff, B., Murray, D.W.: Guided sampling and consensus for motion estimation. In: European conference on computer vision. pp. 82–96. Springer (2002)
25. Welch, G., Bishop, G., et al.: An introduction to the kalman filter (1995)
26. Wu, Y., Tang, F., Li, H.: Image-based camera localization: an overview. Visual Computing for Industry, Biomedicine, and Art **1**(1), 1–13 (2018)
27. Zhang, Z., Scaramuzza, D.: A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7244–7251. IEEE (2018)