Research Article | Vol. 27, No. 17 | 19 Aug 2019 | OPTICS EXPRESS 24481
Optics EXPRESS

PALVO: visual odometry based on panoramic annular lens

HAO CHEN, KAIWEI WANG,* WEIJIAN HU, KAILUN YANG, RUIQI CHENG, XIAO HUANG, AND JIAN BAI

State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, Hangzhou 310027, China
**wangkaiwei@zju.edu.cn*

Abstract: Visual odometry has received a great deal of attention during the past decade. However, being fragile to rapid motion and dynamic scenarios prevents it from practical use. Here, we present PALVO by applying panoramic annular lens to visual odometry, greatly increasing the robustness to both cases. We modify the camera model for PAL and specially design the initialization process based on the essential matrix. Our method estimates the camera's poses through two-stage tracking, meanwhile builds the local map using a probabilistic mapping method based on the Bayesian framework and feature correspondence search along the epipolar curve. Several experiments are performed to verify our algorithm, demonstrating that our algorithm provides an extremely competitive performance in robustness to rapid motion and dynamic scenarios, meanwhile achieves the same level of accuracy as the state-of-the-art visual odometry.

1. Introduction

With the rapid development of robotics and autonomous driving technologies over the past decade, Visual Odometry (VO) has been an area of intense investigation. VO estimates the egomotion of an agent using only the continuous images captured by a single or multiple cameras attached to it, which exhibits apparent discrepancy with other odometers using a code wheel or the like [1]. Depending on the camera setup, VO can be categorized into monocular, stereo, and RGB-D ones. Among them VO with a monocular camera has received considerable attention, for the reason that a monocular camera is more economical and easier to calibrate compared to a stereo (e.g. Stereolabs ZED [2]) or an RGB-D camera (e.g. Intel RealSense [3]). Meanwhile, monocular VO could handle both indoor and outdoor environments, with no consideration for depth aware range, which is a major limiting factor for stereo and RGB-D cameras [4, 5].

A multitude of VO schemes have been presented as of now. Although their pipeline may be distinct from each other, they share a common premise that there must be enough overlaps between two adjacent frames. Only this condition is satisfied, can monocular VO correctly compute the relative motion between two frames. But this is not always the case. For a commonly used perspective camera that has a limited Field-of-View (FoV), fast movement especially rapid rotation will lead to the reduction of overlaps between contiguous frames, resulting in unreliable outputs of VO or even failure in tracking.

On the other hand, current monocular VO mostly has a basic assumption that the target scene must keep stationary during processing, since it's not reliable to model the landmarks corresponding to dynamic objects (such as pedestrians, vehicles, etc.) [6]. But actually the real world doesn't always stay static. Thus in the pipeline of VO, certain algorithms such as RANSAC (RANdom SAmple Consensus) will be applied to eradicate the effects of dynamic objects [7]. These algorithms are effective only if the dynamic components are negligible in the whole field-of-view. However, for a perspective camera with a limited FoV, it's not strange to encounter situations where dynamic objects occupy most of the field-of-view in actual use. In this case, the pose estimation by VO can hardly maintain consistent, or even worse, the VO

#368700 https://doi.org/10.1364/OE.27.024481
Journal © 2019 Received 29 May 2019; revised 24 Jul 2019; accepted 29 Jul 2019; published 13 Aug 2019

Optics EXPRESS

pipeline will get interrupted.

In view of this, we propose to apply Panoramic Annular Lens (PAL) to a monocular visual odometry in this paper, named PALVO. PAL is one of the specific wide angle lenses, which can project 360° FoV onto a planar annular image at one time without rotating the lens, as shown in Fig. 1(a) [8]. A clear benefit of 360° panoramic view lies in that it can ensure enough overlap area between adjacent frames despite fast movement and rapid rotation. Meanwhile it's with extremely low probability that dynamic objects are dominant in a single shot. Thus, the above-mentioned disadvantages caused by the limited FoV will be not a problem for PALVO, greatly improving the robustness of VO to rapid motion and dynamic objects in realistic scenarios.

The contribution of this paper is threefold: 1. We present a modified camera model for PAL based on previous work, which describes the mathematical relationship between the coordinate of a point in 3D (three-dimensional) space and its projection onto the image plane of PAL; 2. A monocular visual odometry for PAL is proposed, in which we specially design initialization progress, tracking method, feature correspondence search along the epipolar curve and depth filter for the PAL camera model; 3. We verify the presented PALVO on both synthetic dataset and real-world dataset collected by a camera with PAL. We analysis the accuracy and robustness of our algorithm, and compare it with existing monocular VO using perspective cameras, demonstrating the superiority of 360° panoramic view.

The rest of this paper is organized as follows. A brief summary of some of the relevant work about visual odometry is presented in Section 2. In Section 3, we introduce the camera model used in this paper. A detailed description of PALVO pipeline is given in Section 4. In Section 5, we perform some experiments to evaluate our algorithm. Finally, conclusions are drawn and directions for future research are discussed in Section 6.

2. Related work

In this section, a brief review of representative works on visual odometry is given. It's notable that visual odometry and visual SLAM (Simultaneous Localization And Mapping) are often discussed together, for the reason that visual SLAM is actually VO with loop closure and global optimization [9]. Thus some research on visual SLAM will be reviewed as well.

2.1. Visual odometry

According to whether explicitly extracting and matching features or not, current existing VO schemes can be mainly divided into feature-based methods which dominated the area for a long time, and direct methods which gained popularity in recent years. The feature-based methods compute the 6 Degree-of-Freedom (DoF) pose by matching salient and repeatable features over frames and minimizing the re-projection error. While direct methods directly exploit pixel intensity values instead of hand-crafted high-dimensional feature descriptors and minimize the photometric error between adjacent frames to estimate the motion.

Feature-based methods MonoSLAM was proposed by Davison et al. in 2007 [10]. It is considered the first visual SLAM. In MonoSLAM, camera motion and 3D position of landmarks are simultaneously estimated using Extended Kalman Filter (EKF) [11]. The drawback of EKF is that computational cost increases in proportion to the size of the environment. Klein et al. proposed PTAM (Parallel Tracking And Mapping) in 2007 [12], which is the first work to introduce the idea of split tracking and mapping into two separate tasks processed in parallel threads. Mapping is based on keyframes, which are processed using Bundle Adjustment (BA) [13]. Compared to EKF, BA yields more accurate results while maintains a reasonable rate of exploration. Built on the main ideas of PTAM, Mur-Artal et al. designed a novel monocular SLAM system named ORB-SLAM in 2014 [14, 15]. ORB-SLAM incorporates a third thread of loop closing besides tracking and local mapping. It can process sequences from indoor and outdoor scenes and achieve impressive accuracy. They then presented a new version of

Optics EXPRESS

ORB-SLAM2 for stereo and RGB-D cameras in addition to monocular cameras in 2017 [16]. It is the current state-of-the-art feature-based SLAM algorithm.

Direct methods Newcombe et al. proposed DTAM (Dense Tracking And Mapping) [17], in which the pose estimation is done by simultaneously creating a dense 3D surface model and using it for dense camera tracking via whole image registration. Engel et al. presented LSD-SLAM (Large-Scale Direct monocular SLAM) in 2014 [18]. LSD-SLAM uses direct image alignment coupled with filtering-based estimation of semi-dense depth maps. It reconstructs consistent, large-scale maps of the environment, which are limited to areas that have intensity gradient. It was also Engel et al. who proposed DSO (Direct Sparse Odometry), which is another leading algorithm of direct methods [19]. Same as LSD-SLAM, points with high intensity are chosen as reconstruction candidates. One of the contributions of DSO has been to incorporate full photometric calibration [20] besides traditionally used geometric calibration, which increases performance compared to the basic "brightness constancy" assumption. Forster et al. proposed SVO (Semi-direct monocular Visual Odometry) in 2014 [21]. The reason why it's called "semi-direct" is that it uses feature-correspondence and minimizes re-projection error (like feature-based methods) while feature correspondence is an implicit result of direct motion estimation rather than of explicit feature extraction and matching (like direct methods). The authors claimed that SVO could produce a quite high frame-rate motion estimate on a consumer laptop.

All the aforementioned works provide high performance under ideal laboratory conditions. However, when it comes to realistic scenarios where the agent with the camera on it moves fast or rotates rapidly, or there are dynamic objects in the environment, the existing VO/SLAM might fail in giving a correct result. The limited FoV camera they used makes them sensitive to these challenging circumstances. In this paper, we take advantage of PAL, which provides large-scale perception of global scene, to tackle this problem.

2.2. VO with wide FoV cameras

Some researchers have recognized the impact of the camera FoV and optics on the quality of motion estimate. Here we review some works on visual odometry or visual SLAM using wide FoV cameras.

Gutierrez et al. developed a real-time visual SLAM system using monocular omnidirectional vision in 2011 [22]. They took advantage of catadioptric cameras by integrating it in the EKF-based SLAM, demonstrating that the system works better with omnidirectional vision since features last longer. In 2018, Matsuki et al. extended DSO and made it applicable to fisheye cameras with a field-of-view well above 180° [23]. They claimed that omnidirectional DSO provides increased accuracy and robustness. In the same year, Wang et al. extended the ORB-SLAM2 framework by applying fisheye cameras and introducing the semi-dense depth map. Lin et al. designed PVO (Panoramic Visual Odometry) [24] for Ricoh Theta V panoramic camera [25], which is a multi-camera system composed of two fisheye lenses and produces 360° FoV with stitch image processing.

Compared to these works with wide FoV imaging systems (fisheye lenses, catadioptric cameras and multi-camera panoramic imaging systems), we propose to apply PAL to visual odometry for the reason that PAL has significant advantages over other panoramic imaging approaches. For instance, PAL is a compact panoramic solution with single sensor, tractable distortion, and concise omnidirectional imaging law. It transforms off-axis light rays from large FoV to approximately paraxial rays. Therefore, aberrations including distortion are reduced so that only a few relay lens are needed [26]. By contrast, the distortion of fisheye lens is hard to control at the margin of the fields, leading to loss of image quality. A multi-camera panoramic imaging system synthesizes panoramic view by stitching images simultaneously captured by several cameras, which can be bulky, expensive and computational costly. Additionally, the projection law for a

Optics EXPRESS

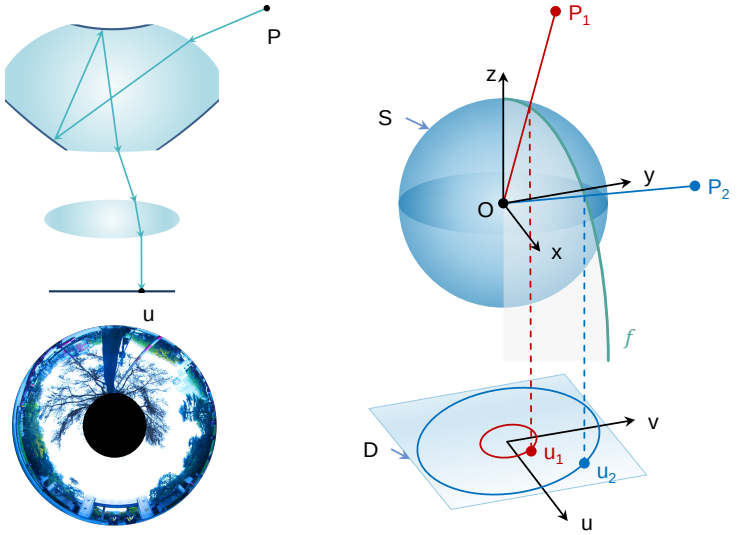

Fig. 1. (a) PAL transforms the cylindrical side view onto a planar annular image. P and u represent the object and image points respectively. (b) Schematic diagram of the PAL camera model. $\mathbf{P}_1$ and $\mathbf{P}_2$ are points in 3D space with projections $\mathbf{u}_1$ and $\mathbf{u}_2$ on image plane D. Without loss of generality, $\mathbf{P}_1$ and $\mathbf{P}_2$ are points at upper and lower margin FoV respectively. The optical center is O, which is also the origin of the O-xyz coordinate system. S represents the unit sphere. f is the polynomial with coefficients to be determined.

multi-camera system can be complicated, and the visual odometry with such a panoramic system must pay attention to synchronization and data fusion between different cameras, which would be an extra burden besides a normal VO procedure and leads to an increase in computing costs. The mirrors used in catadioptric cameras are usually large and irregular, which are hard to fabricate and need high-precision alignment [27]. It prevents this kind of catadioptric systems from broadly use. Given that PAL has the above advantages, we make use of PAL to design a monocular VO in this paper. The proposed PALVO framework processes panoramic view in a single image with a concise omnidirectional projection formula. The compact structure and affordable computing costs guarantee the practical applications in intelligent vehicles and robotics.

3. PAL camera model

The camera model describes the mathematical relationship between the 3D coordinate of a scene point and its projection onto the image plane. Scaramuzza et al. presented an omnidirectional camera model and the corresponding calibration toolbox for catadioptric and fisheye cameras, which was initially described in [28] and later extended in [29]. Built on the main idea of this work, we modify the omnidirectional camera model for catadioptric cameras and make it satisfied for PAL by making the 3D coordinate system mirror symmetric with the original one, in consideration of the fact that every optical ray is reflected twice in a PAL block but only once in a catadioptric system.

As shown in Fig. 1(b), $\mathbf{P}$ is the scene point with coordinate $(x, y, z)^T$, and $\mathbf{u}$ is the projection on image plane with coordinate $(u, v)^T$. As described in [28], $(x, y)^T$ is proportional to $(u, v)^T$. The back-projection (from image plane to 3D space) function π^{-1} can be written as Eq. (1):

$$\mathbf{P} = \pi^{-1}(\mathbf{u}) = \lambda \cdot g(\mathbf{u}), \quad \lambda > 0 \tag{1}$$

Optics EXPRESS

where

$$g(\mathbf{u}) = \begin{bmatrix} u & v & f_b(\rho) \end{bmatrix}^T \tag{2}$$

$$f_b(\rho) = \alpha_0 + \alpha_1 \cdot \rho + \alpha_2 \cdot \rho^2 + \alpha_3 \cdot \rho^3 + \alpha_4 \cdot \rho^4 + \dots \tag{3}$$

$$\rho = \sqrt{u^2 + v^2} \tag{4}$$

It's worth noting that in actual calculation the back-projection $\pi^{-1}(\mathbf{u})$ is not exactly $\mathbf{P}$ but its normalized coordinate $\frac{\mathbf{P}}{||\mathbf{P}||}$ (the projection of $\mathbf{P}$ on the unit sphere), for the reason that from a single view just the bearing vector can be obtained and depth (the distance from $\mathbf{P}$ to the optical center O) is uncertain. Thus, λ is usually calculated as

$$\lambda = \sqrt{\rho^2 + f_b^2(\rho)} \tag{5}$$

The corresponding projection (from 3D space to image plane) π can be modeled with Eq. (6):

$$\mathbf{u} = \pi(\mathbf{P}) = f_p(\theta) \cdot h(\mathbf{P}) \tag{6}$$

where

$$h(\mathbf{P}) = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} & \frac{y}{\sqrt{x^2 + y^2}} \end{bmatrix}^T \tag{7}$$

$$f_p(\theta) = \beta_0 + \beta_1 \cdot \theta + \beta_2 \cdot \theta^2 + \beta_3 \cdot \theta^3 + \beta_4 \cdot \theta^4 + \dots \tag{8}$$

$$\theta = \arctan(\frac{z}{\sqrt{x^2 + y^2}}) \tag{9}$$

Obviously, the polynomials $f_b(\rho)$ and $f_p(\theta)$ are what need to be calibrated, whose coefficients are the calibration parameters to be estimated.

Another thing to note is that we can easily get the derivative of the projection function $\pi(\mathbf{P})$ using chain rules as is shown in Eq. (10), which is a 2×3 Jacobian matrix and plays a role in computing the 6 DoF pose through nonlinear optimization that will be explained in next section.

$$\frac{d\pi(\mathbf{P})}{d\mathbf{P}} = [\frac{\partial \pi}{\partial h}]_{1\times1} \cdot [\frac{\partial h}{\partial \mathbf{P}}]_{2\times3} + [\frac{\partial \pi}{\partial f}]_{2\times1} \cdot [\frac{\partial f}{\partial \theta}]_{1\times1} \cdot [\frac{\partial \theta}{\partial \mathbf{P}}]_{1\times3} \tag{10}$$

where $[\cdot]_{m\times n}$ represents an $m \times n$ matrix. Specifically, $[\cdot]_{1\times1}$ is a scalar.

4. Pipeline

We first give an overview of our system, as illustrated in Fig. 2. For a monocular VO system, a procedure to create an initial map is required, for the reason that depth cannot be obtained from a single view. Thus the initialization module will be executed at first. Then for each new incoming PAL frame, the tracking module will produce a 6 DoF pose by tracking the previous frame and the local map. Meanwhile the depth filter will try to estimate the depth of keypoints and expand the local map, using Bayesian framework and feature correspondence search along the epipolar curve. The local map is composed of a multitude of keypoints with 3D coordinates whose depth is converged after several observations from different frames, and keyframes from which the keypoints can be observed.

Before detailed describing our method, we define some of the notation which will be used in the next subsections. The PAL frame collected at timestamp k is denoted by $\mathbf{I}_k$. For each keypoint

Optics EXPRESS

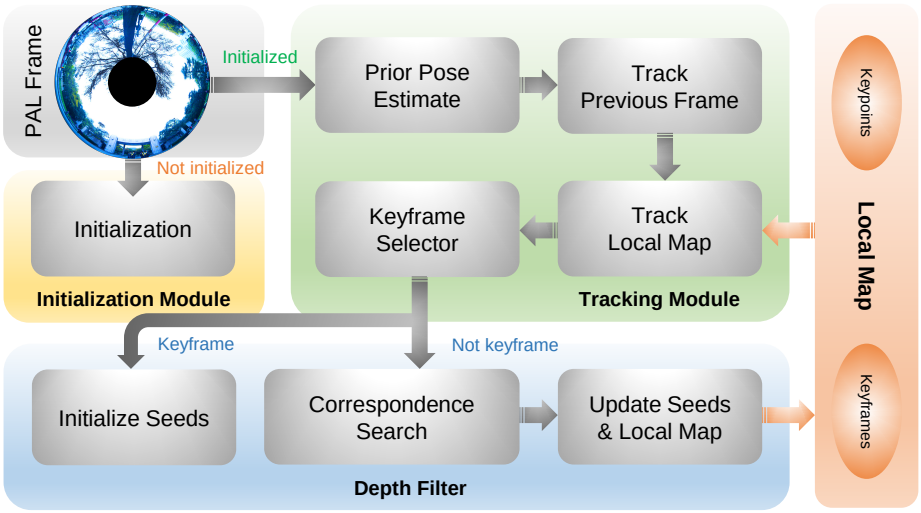

Fig. 2. The pipeline of PALVO.

$\mathbf{P}_i \in \mathbb{R}^3$, it has a corresponding projection $\mathbf{u}_i \in \mathbb{R}^2$ on the image plane, which is the same with Section 3. In addition, the coordinate of $\mathbf{P}_i$ in the camera coordinate system at timestamp k is denoted by $\mathbf{P}_i^k \in \mathbb{R}^3$, and $\mathbf{P}_i^{sk}$ is the normalization of $\mathbf{P}_i^k$, with the meaning of the projection on the unit sphere. $\mathbf{P}_i$, $\mathbf{P}_i^k$ and $\mathbf{P}_i^{sk}$ satisfy the following equation:

$$\mathbf{T}_{k,w} \cdot \mathbf{P}_i = \mathbf{P}_i^k = d \cdot \mathbf{P}_i^{sk} \tag{11}$$

where $d \in \mathcal{R}$ denotes the depth of $\mathbf{P}_i^k$ in the kth frame and equals $||\mathbf{P}_i^k||$. $\mathbf{T}_{k,w} \in SE(3)$ expresses the camera position and orientation (the rigid-body transformation from the world coordinate system to the camera coordinate system at timestamp k). The relative transformation between two consecutive frames $\mathbf{T}_{k,k-1}$ satisfies the following equation:

$$\mathbf{T}_{k,w} = \mathbf{T}_{k,k-1} \cdot \mathbf{T}_{k-1,w} \tag{12}$$

In the next subsections, a detailed description of the three modules (initialization, tracking module and depth filter) will be given.

4.1. *Initialization*

The goal of initialization is to recover the relative pose between two frames and compute the 3D coordinates of keypoints to create an initial map. There are two geometrical models to tackle the problem: the homography and the essential matrix. The former one assumes a planar scene while the latter one is suitable for a non-planar scene. Considering the wide FoV of PAL, the keypoints in a frame are almost impossible to be coplanar. Thus we apply essential matrix to reconstruct the structure from motion.

The essential matrix $\mathbf{E}$ satisfies the following condition:

$$\mathbf{P}_i^{2T} \cdot \mathbf{E} \cdot \mathbf{P}_i^1 = 0 \tag{13}$$

Applying Eq. (11), the equation can be rewritten as

$$d_2 \cdot d_1 \cdot \mathbf{P}_i^{s2T} \cdot \mathbf{E} \cdot \mathbf{P}_i^{s1} = 0 \quad \Rightarrow \quad \mathbf{P}_i^{s2T} \cdot \mathbf{E} \cdot \mathbf{P}_i^{s1} = 0 \tag{14}$$

Optics EXPRESS

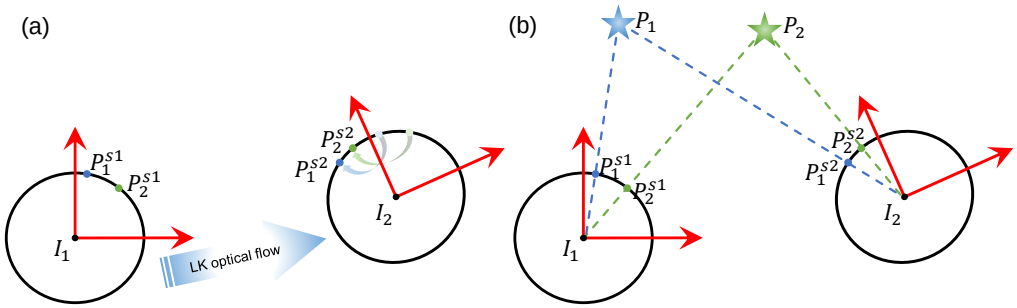

Fig. 3. The initialization module. (a) Initial feature correspondences are determined using Lucas-Kanade optical flow, and the essential matrix is used to compute the relative motion. (b) By triangulating the corresponding keypoints, an initial map can be created.

where $\mathbf{P}_i^{s1}$ and $\mathbf{P}_i^{s2}$ can be obtained by back-projection using the PAL camera model mentioned in Section 3 and then the essential matrix is computable using the eight-point algorithm with a RANSAC scheme [30]. A total of four motion hypotheses can be retrieved with the singular value decomposition [31]. We triangulate the four solutions and score them based on the number of points with positive depth in both cameras, as is shown in Fig. 3(b). The solution with highest score is selected as the winner. The initialization is considered successful only when the highest score is more than 5 times the second highest one, meanwhile the number of triangulated keypoints is greater than 100. Otherwise the initialization is judged as invalid and will restart, in order to avoid initializing a corrupted map.

It's noteworthy that to find initial correspondences, we extract features on the reference frame (the first frame) and track them onto the second one using Lucas-Kanade optical flow [32], without computing feature descriptors and explicit matching (see Fig. 3(a)). The choice of feature can be varied, e.g. SURF keypoints [33], FAST corners [34], or points with large gradient in the image used in DSO [19]. In this paper, we make use of the oriented FAST corners, which has ever been used in [16].

4.2. *Tracking module*

4.2.1. Prior pose estimate

In the tracking module, each new frame acquired from the PAL camera is assigned with a prior pose estimate, which is generated from the uniform motion model reasonably. This initial guess of relative camera motion $\widehat{\mathbf{T}}_{k,k-1}$ can be expressed as

$$\widehat{\mathbf{T}}_{k,k-1} = \mathbf{T}_{k-1,k-2} \tag{15}$$

Then a two-stage tracking process (i.e. tracking the previous frame and then tracking the local map) is performed to refine the pose estimate.

4.2.2. Track previous frame

Taking ideas from SVO [21], we firstly update the relative camera pose and obtain feature correspondences by tracking the previous frame using direct methods. The ith keypoint $\mathbf{P}_i$ with projection $\mathbf{u}_i$ in the previous frame $\mathbf{I}_{k-1}$ is projected into the current frame according to the frame's prior pose estimate, and the intensity residual is calculated as

$$\delta\mathbf{I}(\mathbf{u}_i, \widehat{\mathbf{T}}_{k,k-1}) = \mathbf{I}_k(\mathbf{u}_i^{'}) - \mathbf{I}_{k-1}(\mathbf{u}_i) \tag{16}$$

where

$$\mathbf{u}_i^{'} = \pi(\widehat{\mathbf{T}}_{k,k-1} \cdot d_{u_i} \cdot \pi^{-1}(\mathbf{u}_i)) \tag{17}$$

Optics EXPRESS

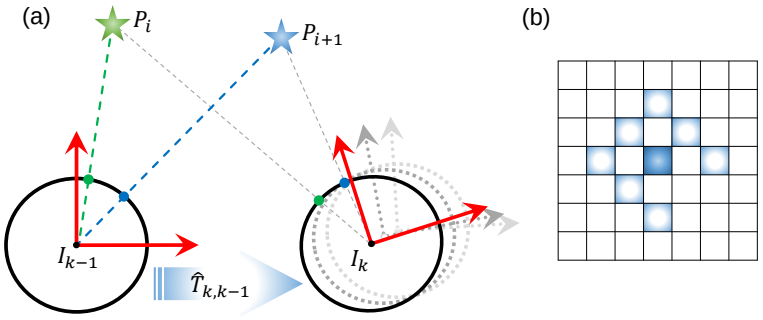

Fig. 4. (a) Track previous frame. We seek to find the relative motion between two consecutive frames that minimizes the photometric error. The colorful dash lines mean affirmed projection, while the gray ones mean undetermined projection because the camera pose is to be estimated. The following Figs. 5 and 6 are the same. (b) Residual pattern. Pattern $\mathcal{N}_{u_i}$ used for photometric error computation [19].

We define the photometric error of the point $\mathbf{u}_i$ as the intensity residuals over a small neighbourhood of pixels that are denoted by $\mathcal{N}_{u_i}$ (see Eq. (18)). For the sake of computational efficiency and providing sufficient information, we select 8 pixels arranged in a specific pattern shown in Fig. 4(b) for error computation according to the conclusion in [19].

$$\mathbf{E}_{u_i}(\widehat{\mathbf{T}}_{k,k-1}) = \frac{1}{2} \sum_{\mathbf{u}_i \in \mathcal{N}_{u_i}} \left\| \delta(\mathbf{u}_i, \widehat{\mathbf{T}}_{k,k-1}) \right\|^2 \tag{18}$$

As shown in Fig. 4(a), an iterative Gauss-Newton procedure is taken to estimate a $\widehat{\mathbf{T}}_{k,k-1}$ that minimizes the photometric error over all the tracked keypoints:

$$\widehat{\mathbf{T}}_{k,k-1} = \arg \min_{\widehat{\mathbf{T}}_{k,k-1}} \sum_i \mathbf{E}_{u_i}(\widehat{\mathbf{T}}_{k,k-1}) \tag{19}$$

In order to determine the Gauss-Newton step for an iteration, the Jacobian matrix associated with each point is estimated by chain rules as

$$\mathbf{J}_{u_i} = \frac{\partial \mathbf{I}_k}{\partial \mathbf{u}_i'} \cdot \frac{\partial \mathbf{u}_i'}{\partial \mathbf{P}_i^k} \cdot \frac{\partial \mathbf{P}_i^k}{\partial \widehat{\mathbf{T}}_{k,k-1}} \tag{20}$$

where

$$\mathbf{P}_i^k = \widehat{\mathbf{T}}_{k,k-1} \cdot d_{u_i} \cdot \pi^{-1}(\mathbf{u}_i) \tag{21}$$

and $\frac{\partial \mathbf{u}_i'}{\partial \mathbf{P}_i^k}$ is exactly the derivative of the projection function $\pi(\cdot)$, which is detailed described in Eq. (10).

4.2.3. Track local map

With the estimated $\widehat{\mathbf{T}}_{k,k-1}$ and previous camera motion, an initial guess of the current camera pose can be calculated as

$$\widehat{\mathbf{T}}_{k,w} = \widehat{\mathbf{T}}_{k,k-1} \cdot \mathbf{T}_{k-1,w} \tag{22}$$

But only tracking the previous frame can be inaccurate due to accumulated drift. Thus the alignment with respect to the local map is necessary.

Optics EXPRESS

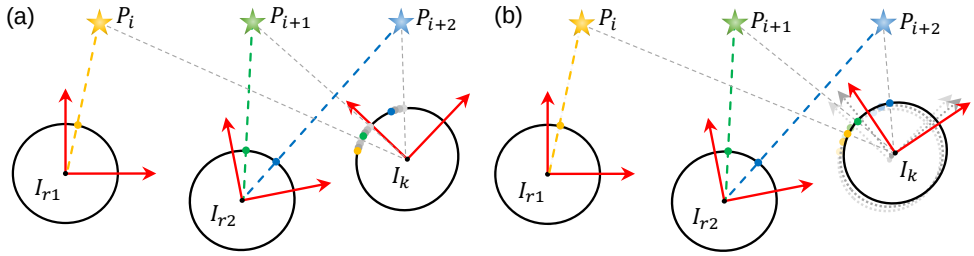

Fig. 5. Track local map. (a) Keypoints of the local map that are visible from the current frame are projected onto the image and then the features get aligned. (b) The pose and structure are optimized to minimize the re-projection error.

As Fig. 5(a) shows, keypoints which can be observed from the estimated camera pose are projected into the current frame, which are denoted by $\widehat{\mathbf{u}}_i$, giving the initial correspondences between 2D features in the current frame and 3D keypoints in the local map. Then the position of 2D features is optimized to minimize the photometric error between the estimated $\widehat{\mathbf{u}}_i$ and their observations in earlier keyframes, which are regarded as reference frames (denoted by $\mathbf{I}_r$), as is shown in Eq. (23). Similarly, we also define the photometric error as the intensity residuals over the pattern shown in Fig. 4(b). But considering differences in viewpoint between the current frame and the reference frames, homography is used during error computation.

$$\mathbf{u}_i = \arg\min_{\widehat{\mathbf{u}}_i} \frac{1}{2} \sum_{\widehat{\mathbf{u}}_i \in \mathcal{N}_{u_i}} \left\| \mathbf{I}_k(\widehat{\mathbf{u}}_i) - \mathbf{I}_r(\pi(\mathbf{H} \cdot \pi^{-1}(\widehat{\mathbf{u}}_i))) \right\|^2 \tag{23}$$

where $\mathbf{H}$ denotes the homography matrix.

Once the feature correspondences are determined, we optimize the camera pose to minimize the re-projection error, as shown in Fig. 5(b) and Eq. (24). The optimization result is treated as the exact pose of the current frame.

$$\mathbf{T}_{k,w} = \arg\min_{\widehat{\mathbf{T}}_{k,w}} \frac{1}{2} \sum_i \left\| \pi(\widehat{\mathbf{T}}_{k,w} \cdot \mathbf{P}_i) - \mathbf{u}_i \right\|^2 \tag{24}$$

4.2.4. Keyframe selector

After pose estimation, the keyframe selector should determine whether the current frame is a keyframe or not. A keyframe contributes to expanding the local map by initializing new keypoints with unknown depth named seeds, and a non-keyframe updates the depth of seeds using the depth filter.

A frame is selected as a keyframe in any of the following circumstances:

- Over 30% of the keypoints in the previous frame get lost;
- Fewer than 50 keypoints in the local map are tracked;
- No keyframe is generated in the past 10 frames;
- The depth filter thread is idle.

4.3. Depth filter

Once a frame is registered as a keyframe, seeds are created with an initial depth guess with a large variance, which are modeled with Gaussian distribution as $\mathcal{N}(\widehat{d}, \sigma^2)$. And when a new non-keyframe is added to the depth filter, stereo matching based on correspondence search along

Optics EXPRESS

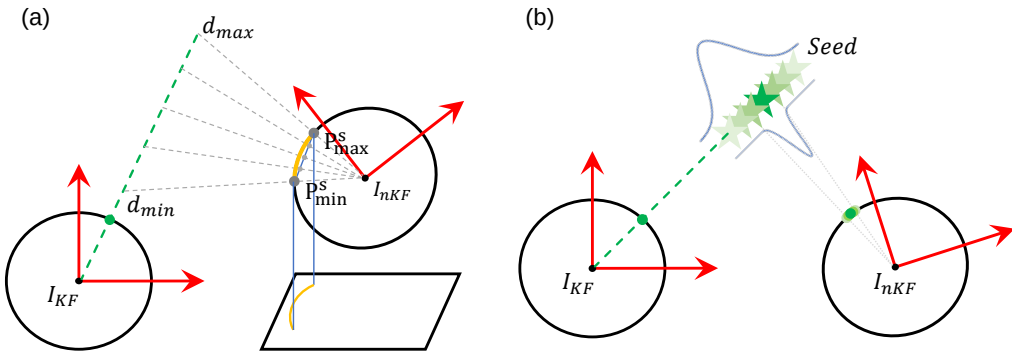

Fig. 6. (a) Feature correspondence search along the epipolar curve. (b) The depth estimate is updated in a Bayesian framework.

the epipolar is performed to update the depth of seeds. Different to a perspective camera, the epipolar in PAL camera model is not a straight line but a "curve". Following the work of [35], we apply a simple but efficient way to perform stereo on the PAL camera model.

As is shown in Fig. 6(a), $\mathbf{P}^s_{min}$ and $\mathbf{P}^s_{max}$ are projections of the seed with minimum and maximum depth estimate on the unit sphere respectively. A straight line segment is built between $\mathbf{P}^s_{min}$ and $\mathbf{P}^s_{max}$, which can be expressed by

$$\mathbf{P}_L(\alpha) = \alpha \cdot \mathbf{P}^s_{max} + (1-\alpha) \cdot \mathbf{P}^s_{min}, \quad \alpha \in [0,1] \tag{25}$$

We sample this line segment and project $\mathbf{P}_L(\alpha)$ onto the image plane to perform matching using the cost of SSD (Sum of Squared Differences).

In order to enforce that the search step size is less than 1 pixel, we estimate the number of samples using an empirical formula:

$$N_{sample} = \frac{\pi}{2} \cdot \left\| \pi(\mathbf{P}^s_{max}) - \pi(\mathbf{P}^s_{min}) \right\| \tag{26}$$

As shown in Fig. 6(b), according to the correspondence search result, the depth estimate can be updated with the triangulated depth d_{tri} in a Bayesian framework as follows:

$$\widehat{d}_{new} = \frac{\sigma^2_{old} \cdot d_{tri} + \sigma^2_{tri} \cdot d_{old}}{\sigma^2_{old} + \sigma^2_{tri}} \tag{27}$$

$$\sigma^2_{new} = \frac{\sigma^2_{old} \cdot \sigma^2_{tri}}{\sigma^2_{old} + \sigma^2_{tri}} \tag{28}$$

Once the variance of a seed is decreased to 0.5% of its initial variance, it's considered as converged and added to the local map. Otherwise a seed is abandoned if it fails in stereo matching with 10 consecutive non-keyframes.

In summary, the pipeline of PALVO is accomplished by initialization, two-stage tracking and depth filter, which is specially designed for panoramic view. It produces 6 DoF poses and simultaneously maintains a local map consisting of keyframes and keypoints with 3D coordinates.

5. Experiments

5.1. Experimental setup

We verify our algorithm on both real-world and synthetic dataset, and compare the performance with SVO and monocular ORB-SLAM2, which are the state-of-the-art visual odometry/SLAM

Optics EXPRESS

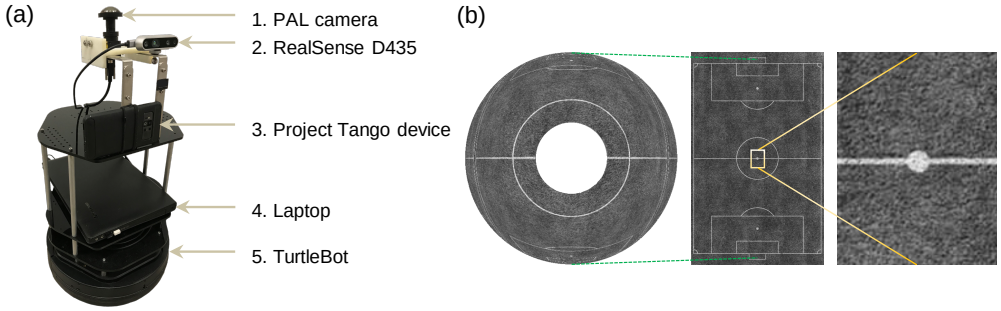

Fig. 7. (a) Experimental platform. The real-world dataset is collected using the robot of TurtleBot, with a PAL camera, a RealSense D435, a project tango device and a laptop on it. (b) The images of synthesis dataset are generated by a virtual camera which gives an aerial view to a football field. Left: the PAL image. Medium: the football field. Right: the perspective image.

that take use of a perspective camera. Each dataset is composed of PAL image sequences and corresponding perspective image sequences, which are captured or synthesized in the same trajectory. The real-world dataset is collected using the robot of TurtleBot, as shown in Fig. 7(a). The PAL camera placed on top of the robot provides PAL sequences, and perspective sequences are gathered from the color stream of RealSense D435, with a 77° field-of-view in diagonal. Meanwhile, we adopt a Project Tango device [36] to produce a reference trajectory. As for the synthesis dataset, we assume that the camera is in the air giving an aerial view of the ground. The virtual camera moves in designed trajectories and the images are generated according to real camera models, which are obtained by calibrating the cameras used in collecting real-world dataset (both PAL and RealSense's RGB camera). The designed trajectories are regarded as the ground truth in evaluating the accuracy of the algorithm.

It needs to be emphasized that we specifically collect image sequences in the following two circumstances: 1. The TurtleBot moves or rotates with a high velocity. 2. There are moving objects in the environment. These two application scenarios could be daunting challenges for traditional VO with a monocular perspective camera. By evaluating the performance of our algorithm in these scenarios, the competitive advantage of PALVO is illustrated.

5.2. Accuracy test

Firstly we perform the accuracy test on the synthesis dataset. As is mentioned above, we assume a virtual camera in the air giving an aerial view to a football pitch, which is shown in Fig. 7(b). As we can see, from the synthetic PAL image almost the entire football field can be seen except for the blind area of PAL, while for the perspective image one can see just a narrow area. We designed 5 different trajectories and run the three algorithms (PALVO, SVO and monocular ORB-SLAM2) on each sequences, computing the Root Mean Square Error (RMSE) over all time indicies of the translational component. Since a camera is only a bearing-only sensor, it's impossible to obtain the scale of the map through a VO pipeline. Thus we align the scale of the first 10 measurements with the ground truth, as SVO did in [21].

Secondly, we also compare the accuracy of the proposed PALVO with the other two algorithms on a real-world dataset. Since the ground truth of 6 DoF poses for each frame in real-world image sequences cannot be obtained conveniently, we utilize loop closure error as a criterion for accuracy. Specifically, we collect 5 image sequences along different paths that start and end at the same point. The residual between the starting and ending point of the trajectory produced by VO is calculated, and the ratio of the residual to the entire length of trajectory indicates loop

Optics EXPRESS

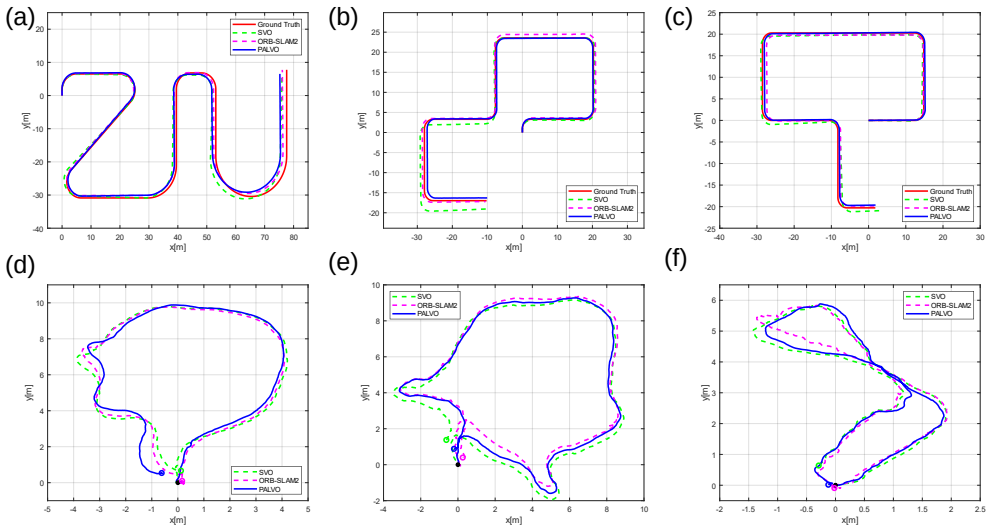

Fig. 8. The trajectories produced by PALVO, ORB-SLAM2 and SVO on the synthesis image sequences s1 (a), s4 (b), s5 (c) and on the real-world image sequences with loop closure r3 (d), r4 (e) and r5 (f). In (d)-(f), the starting point is indicated by a black dot, and the ending points of trajectories produced by different algorithms are represented by circles in different color.

Table 1. **Accuracy test results.**

	Frame Rate	Translational RMSE (m)					Loop Closure Error (%)				
Method	FPS	s1	s2	s3	s4	s5	r1	r2	r3	r4	r5
ORB-SLAM2	37.4	0.9454	1.3281	1.4212	1.2256	0.3957	0.8364	1.2000	0.5425	1.2681	0.6779
SVO	423.9	1.5106	2.5244	2.3499	1.3900	1.1124	1.4276	2.0067	2.8455	3.5414	1.7109
PALVO	251.6	1.4046	2.2415	2.3963	0.4988	0.6739	1.9326	1.5893	2.9858	2.6105	0.9527

closure error. By the way, considering that the running time and computational efficiency is important to VO, the FPS (frame per second) values are also recorded during the accuracy test.

Fig. 8(a)-8(c) give three examples of the designed trajectories in the synthesis dataset (s1, s4 and s5), and Fig. 8(d)-8(f) show the trajectories of three sequences in the real-world dataset (r3, r4 and r5). All quantitative results of the accuracy test are shown in Table 1. It's indicated that on most synthesis sequences (s1, s2, s3, s5) the accuracy of PALVO is a little inferior to ORB-SLAM2, but parallel to SVO (s1, s2, s3) or better than it (s5). On the 4th sequence (s4), PALVO performs the best. As for the real-world dataset, ORB-SLAM2 achieves the best performance on all of the sequences. PALVO is inferior to SVO on 2 sequences (r1 and r3) and better to it on the other 3 sequences (r2, r4 and r5). The experiment results indicate that our algorithm has achieved the same level of accuracy as SVO, which is a state-of-the-art VO scheme, but is inferior to the up-to-date SLAM system. There are two main reasons for this result. One is that ORB-SLAM2 is a full SLAM system with global optimization, which reduces the accumulated drift. The other reason lies in the insufficient angular resolution of PAL, restricting the precision of feature correspondence. Regarding the defect of lower angular resolution, we perform sub-pixel processing during feature correspondence search, which overcomes this disadvantage to some extent and allows our algorithm to maintain approximately the same accuracy as SVO.

As for the running time, as one can see in Table 1, the proposed PALVO is capable to process

Optics EXPRESS

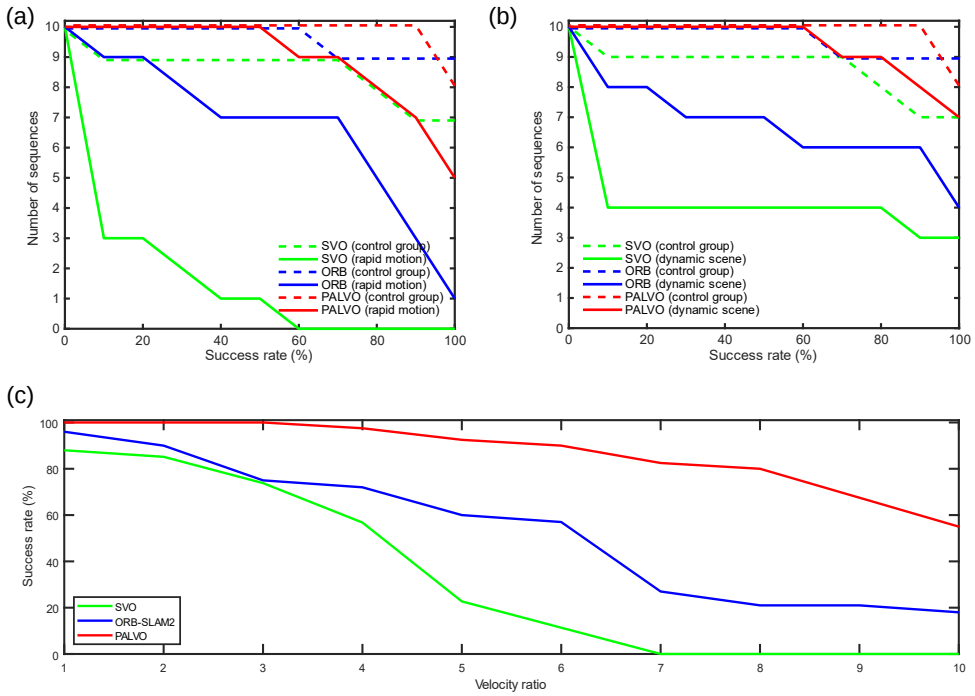

Fig. 9. Experiment results. We run the three methods (PALVO, SVO and monocular ORB-SLAM2) 10 times on each sequence and draw diagrams of number of sequences with a success rate greater than N% as a function of N. The dash lines represent results of the control group (slow motion and static scenarios), while the real lines stand for results of the experimental groups with rapid motion (a) and dynamic scenarios (b). The higher the curve, the more robust the algorithm. Moreover, we gradually increase the velocity from the basic speed to 10 times faster, and run the three algorithms 40 times at each velocity. The relationship between success rate and the velocity ratio is shown in (c).

the frames at 251.6 FPS, which is much faster than ORB-SLAM2 (37.4 FPS) but slower than SVO (423.9 FPS). Considering the significant improvement in robustness, which will be illustrated in next experiments, meanwhile maintaining the same level of accuracy as SVO and a much faster running speed than ORB-SLAM2, our algorithm provides an extremely competitive performance.

5.3. Rapid motion

Several experiments are performed to verify the positive effect of PAL on robustness to rapid motion, which challenges existing VO methods a lot. For a fair comparison, we set a control group consist of sequences with slow motion and static scenarios, which is an ideal operating environment for a VO algorithm. While the experimental group is set with static scenarios but rapid motion. In our experiment, the speed of slow motion is set as 0.1m/s linear velocity and 0.314 rad/s angular velocity, while for rapid motion they are three times of these two values.

Each group contains 10 different sequences. In order to prevent accidental errors, we run the three methods (PALVO, SVO and monocular ORB-SLAM2) 10 times on each sequence and compare their mean performance. The statistics of success rate is used as the evaluation criteria for robustness. For the sake of brevity and clarity, we draw diagrams of number of sequences with a success rate greater than N% as a function of N. For example, on the red dash curve which represents PALVO (control group) in Fig. 9(a), when the success rate is 90%, the number of

Optics EXPRESS

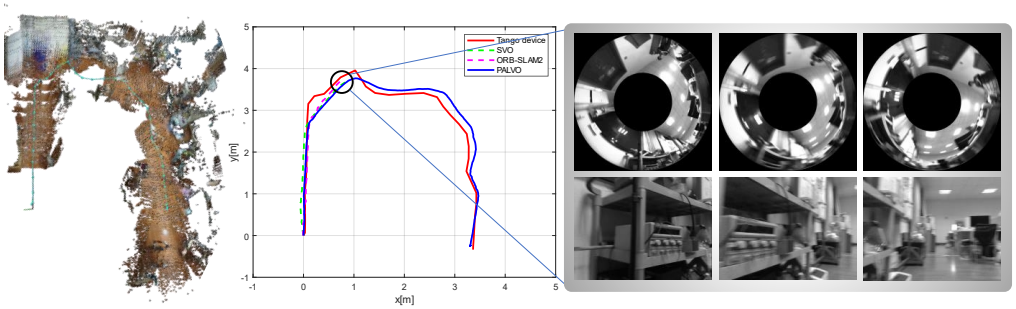

Fig. 10. Trajectories produced by PALVO, ORB-SLAM2, SVO and Tango device. Both SVO and ORB-SLAM2 failed in tracking when turning at a fast angular velocity. The picture on the left shows the point cloud produced by Tango device.

sequences is equal to 10, meaning that the success rate of PALVO is greater than 90% on all of the 10 sequences. Obviously, the higher the curve, the more robust the algorithm.

The results of three algorithms on the control group and on the sequences with rapid motion are shown in Fig. 9(a). As we can see, on the control group, the success rate of PALVO is greater than 90% on all of the 10 sequences, while ORB-SLAM2 and SVO have a same success rate on 9 and 7 sequences respectively. PALVO and ORB-SLAM2 have almost identical performance, with a slight adavntage over SVO. For the case of rapid motion, PALVO still achieves a satisfactory result, with a success rate of greater than 90% for 7 sequences. But ORB-SLAM2 performs not so well in this circumstance, with a success rate of greater than 90% for just 3 sequences, as for SVO the number is even 0, which performs the worst.

In order to further demonstrate the improvement of our algorithm, another experiment is conducted. We set a base group with slow motion and static scenarios, and the basic speed is set as 0.05 m/s linear velocity and 0.157 rad/s angular velocity. Then the speed is gradually increased to twice, 3 times until up to 10 times of the basic value. A total of 10 image sequences are collected at these different velocities. We run three algorithms 40 times on each of the sequences and compare their success rate. The results are shown in Fig. 9(c). As we can see, with the velocity increasing, the success rate of ORB-SLAM2 and SVO shows a significant decline, while the proposed PALVO maintains a high level of success rate. It's no doubt that PAL improves the robustness of VO to rapid motion significantly.

Fig. 10 shows a sequence on which both SVO and ORB-SLAM2 get failed. It can be seen that the trajectory estimated by PALVO is almost identical to the trajectory given by Project Tango device. But SVO and ORB-SLAM2 both failed in tracking when turning at a high angular velocity. The rapid rotation results in drastic reduction of the overlap between consecutive frames, leading to the unsuccessful run. But for PAL, thanks to the panoramic view, quite a number of objects remain in the FoV during rotation, ensuring the robustness of our algorithm to this circumstance.

5.4. Dynamic scenarios

We also study the robustness of PALVO in dynamic scenarios and compare it with SVO and ORB-SLAM2. The control group with slow motion and static scenarios in Section 5.3 is again taken for comparison, and the experimental group is set with slow motion but dynamic scenarios. A total of 10 image sequences were taken when there were pedestrians walking in front of the robot, leading to most FoV occupied by dynamic objects (pedestrians). Similarly, we run the three algorithms 10 times on each sequences and calculate statistical results.

The running results of three algorithms are shown in Fig. 9(b). As we can see, PALVO is least

Optics EXPRESS

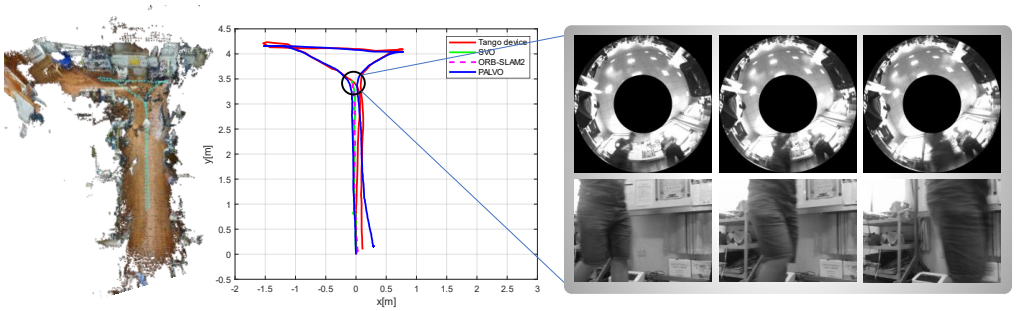

Fig. 11. Trajectories produced by PALVO, ORB-SLAM2, SVO and Tango device. Both SVO and ORB-SLAM2 got interrupted when there's a pedestrian walking in front of the camera. The picture on the left shows the point cloud produced by Tango device.

affected and maintains a high success rate on most sequences, but for SVO and ORB-SLAM2, the opposite is true. Only on 3 sequences SVO has a success rate greater than 90%. Compared to the control group, it's safe to say dynamic scenarios bring a disastrous effect to VO with a perspective camera, but a very slight impact on PALVO.

An unsuccessful run of SVO and ORB-SLAM2 is shown in Fig. 11. It can be seen that the walking pedestrian interrupts the tracking process of SVO and ORB-SLAM2, but has no impact on PALVO because of low proportion in the PAL's FoV, of which the trajectory maintains high consistency with Project Tango device.

5.5. Field test

For the sake of further evaluating our algorithm and validating its effectiveness and reliability in actual use, field tests are conducted in the outdoor area. We utilize a remote control car equipped with a PAL camera on it and drive the car along two paths at a velocity of about 3m/s. Each trajectory is around 150 meters long. There exist 360° sharp turns during driving, meanwhile there are plenty of dynamic objects such as pedestrians and vehicles, which is often the situation a visual odometry algorithm has to deal with when actually applied to intelligent vehicles.

The running results of our algorithm are shown in Fig. 12. The two trajectories are highlighted in different color, one in blue and the other in red. It can be seen that even in the case of a fast U-turn (A) and a vehicle passing by (B), our algorithm runs successfully and produces correct trajectories. The experiment results further prove the robustness of the proposed PALVO to rapid motion and dynamic objects, and partially demonstrate the practicality of our algorithm.

6. Conclusion and future work

In this paper, we propose PALVO by applying panoramic annular lens to visual odometry. Based on the PAL camera model, we specially design the initialization module with essential matrix, two-stage tracking with prior pose estimate, and depth filter with feature correspondence search along the epipolar curve. Experiments demonstrate that our method is extremely robust to rapid motion and dynamic scenarios, which have been daunting challenges for a normal VO. The gain in robustness is due to the wide FoV of PAL, which brings a larger overlap between adjacent frames and smaller proportion of moving objects in the whole FoV. Our algorithm achieves the same level of accuracy as the state-of-the-art VO while brings the dramatic improvement in robustness, greatly enhancing the practicality in actual use such as robot localization and autonomous driving. Additionally, our visual odometry framework can be easily extended to catadioptric and fisheye cameras, since they share the same projection relationship described by the camera model we used, except for some modifications.

Optics EXPRESS

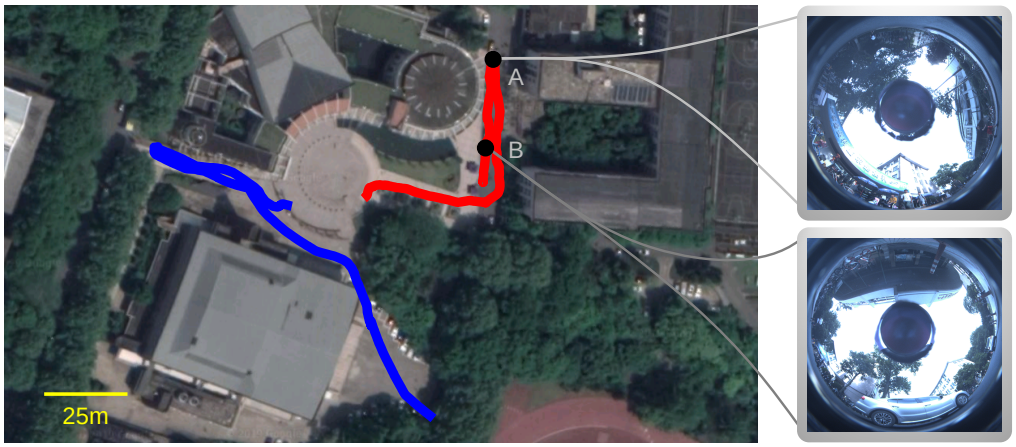

Fig. 12. Trajectories produced by PALVO in field tests. There exists a fast U-turn at A, and there is a car passing by at B. Despite this, PALVO runs successfully and produces correct trajectories.

In the future, we aim to decrease the accumulated drift by introducing loop closure, generating a full SLAM system [37]. Additionally, we have the intention to develop semantic SLAM based on PAL by incorporating robust panoramic annular semantic segmentation [38–40].

Funding

Zhejiang Provincial Public Fund (No. 2016C33136); State Key Laboratory of Modern Optical Instrumentation (501100011312).

References

1. D. Scaramuzza and F. Fraundorfer, "Tutorial: Visual odometry," IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011).
2. Stereolabs, "ZED Stereo camera," https://www.stereolabs.com/zed/.
3. L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (IEEE, 2017), pp. 1–10.
4. H. Chen, K. Wang, and K. Yang, "Improving realsense by fusing color stereo vision and infrared stereo vision for the visually impaired," in Proceedings of International Conference on Information Science and System (ACM, 2018), pp. 142–146.
5. K. Yang, K. Wang, H. Chen, and J. Bai, "Reducing the minimum range of a rgb-depth sensor to aid navigation in visually impaired individuals," Appl. Optics **57**(11), 2809–2819 (2018).
6. M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," ACM Comput. Surv. **51**(2), 37:1–37:36 (2018).
7. W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in Proceedings of IEEE International Symposium on Mixed and Augmented Reality (IEEE, 2013), pp. 209–218.
8. Y. Luo, X. Huang, J. Bai, and R. Liang, "Compact polarization-based dual-view panoramic lens," Appl. Optics **56**(22), 6283–6287 (2017).
9. T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," IPSJ Transactions on Computer Vision and Applications **9**(1), 16 (2017).
10. A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007).
11. G. Welch, G. Bishop, "*An introduction to the Kalman Filter*," (University of North Carolina at Chapel Hill, 1995).
12. G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (IEEE, 2007), pp. 1–10.
13. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," in Proceedings of International Workshop on Vision Algorithms, B. Triggs, P.F. McLauchlan, R.I Hartley and A.W. Fitzgibbon, ed. (Springer, 2000), pp. 298–372.
14. R. Mur-Artal and J. D. Tardós, "ORB-SLAM: Tracking and mapping recognizable features," presented at Robotics: Science and Systems (RSS) Workshop on Multi View Geometry in Robotics, Berkeley, USA, July 2014.

Optics EXPRESS

15. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," IEEE Trans. Robot. **31**(5), 1147–1163 (2015).
16. R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Trans. Robot. **33**(5), 1255–1262 (2017).
17. R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in Proceedings of International Conference on Computer Vision (IEEE, 2011), pp. 2320–2327.
18. J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in Proceedings of European Conference on Computer Vision, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars ed. (Springer, 2014), pp. 834–849.
19. J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2017).
20. P. Bergmann, R. Wang, and D. Cremers, "Online photometric calibration of auto exposure video for realtime visual odometry and slam," IEEE Robotics and Automation Letters **3**(2), 627–634 (2018).
21. C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in Proceedings of IEEE International Conference on Robotics and Automation (IEEE, 2014), pp. 15–22.
22. D. Gutierrez, A. Rituerto, J. Montiel, and J. J. Guerrero, "Adapting a real-time monocular visual slam from conventional to omnidirectional cameras," in Proceedings of IEEE International Conference on Computer Vision Workshops (IEEE, 2011), pp. 343–350.
23. H. Matsuki, L. von Stumberg, V. Usenko, J. Stückler, and D. Cremers, "Omnidirectional DSO: Direct sparse odometry with fisheye cameras," IEEE Robotics and Automation Letters **3**(4), 3693–3700 (2018).
24. M. Lin, Q. Cao, and H. Zhang, "PVO: Panoramic visual odometry," in Proceedings of IEEE International Conference on Advanced Robotics and Mechatronics (IEEE, 2018), pp. 491–496.
25. Ricoh Company, "RICOH THETA V," https://theta360.com/en/about/theta/v.html.
26. Z. Huang, J. Bai, T. X. Lu, and X. Y. Hou, "Stray light analysis and suppression of panoramic annular lens," Opt. Express **21**(9), 10810–10820 (2013).
27. Z. Huang, J. Bai, and X. Y. Hou, "Design of panoramic stereo imaging with single optical system," Opt. Express **20**(6), 6085–6096 (2012).
28. D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in Proceedings of IEEE International Conference on Computer Vision Systems (IEEE, 2006), pp. 45–55.
29. D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE, 2007), pp. 5695-5701.
30. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," Nature **293**(5828), 133 (1981).
31. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision* (Cambridge university press, 2003).
32. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of the 7th International Joint Conference on Artificial Intelligence, (Morgan Kaufmann Publishers Inc., 1981), pp. 674–679.
33. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in Proceedings of European Conference on Computer Vision, A. Leonardis, H. Bischof and A. Pinz, ed. (Springer, 2006), pp. 404–417.
34. M. Trajković and M. Hedley, "Fast corner detection," Image Vis. Comput. **16**(2), 75–87 (1998).
35. D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE, 2015), pp. 141–148.
36. E. Marder-Eppstein, "Project tango," in Proceedings of ACM SIGGRAPH 2016 Real-Time Live! (ACM, 2016), pp. 40:25–40:25.
37. R. Cheng, K. Wang, S. Lin, W. Hu, K. Yang, X. Huang, H. Li, D. Sun, and J. Bai, "Panoramic annular localizer: Tackling the variation challenges of outdoor localization using panoramic annular images and active deep descriptors," arXiv preprint arXiv:1905.05425 (2019).
38. H. Chen, K. Wang, W. Hu, and L. Fei, "SORB: Improve ORB feature matching by semantic segmentation," Proc. SPIE 10799, 107990Z–107990Z-7 (2018).
39. K. Yang, L. M. Bergasa, E. Romera, and K. Wang, "Robustifying semantic cognition of traversability across wearable rgb-depth cameras," Appl. Optics **58**(12), 3141–3155 (2019).
40. K. Yang, X. Hu, L. M. Bergasa, E. Romera, X. Huang, D. Sun, and K. Wang, "Can we PASS beyond the field of view? panoramic annular semantic segmentation for real-world surrounding perception," in Proceedings of IEEE Intelligent Vehicles Symposium (IEEE, 2019), pp. 374–381.