2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

360VO: Visual Odometry Using A Single 360 Camera

Huajian Huang and Sai-Kit Yeung

***Abstract*— In this paper, we propose a novel direct visual odometry algorithm to take the advantage of a 360-degree camera for robust localization and mapping. Our system extends direct sparse odometry by using a spherical camera model to process equirectangular images without rectification to attain omnidirectional perception. After adapting mapping and optimization algorithms to the new model, camera parameters, including intrinsic and extrinsic parameters, and 3D mapping can be jointly optimized within the local sliding window. In addition, we evaluate the proposed algorithm using both real world and large-scale simulated scenes for qualitative and quantitative validations. The extensive experiments indicate that our system achieves start of the art results.**

I. INTRODUCTION

Visual odometry (VO) and visual simultaneous localization and mapping (VSLAM) are fundamental problems that seek to exploit visual information to estimate agents' poses with respect to the observed environments. Realtime and robust systems play an important role in various robotic applications, such as autonomous vehicles, robot navigation, and augmented reality.

An effective approach to enhance system robustness is to increase the field of view (FOV). As the FOV increases, an image can capture more texture and covisible regions among sequential frames are expanded as well. It enables the system to acquire sufficient feature correspondence even in structure-less environments. In addition, as the proportion of dynamic objects in each frame decreases, it is easier for the system to discard outliers improving the accuracy of pose estimations. In the meanwhile, mature lens manufacturing technology can produce high-quality but affordable camera lenses with more than 180-degree FOV. Consequently, we can realize a fully omnidirectional perception and capture 360 degree horizontal and 180 degree vertical information via only two lenses attached back to back, as Fig. 1 (top) shows. The cost of 360 camera has been reduced and its calibration process is simplified, while commercial products are becoming more and more popular and accessible, such as Insta360, RICOH THETA, etc.

However, when the FOV of camera increases even above 180 degrees, features distortion becomes non-linear and the images cannot be properly rectified into perspective images. As a result, the standard systems [1]–[5] using the pinhole camera model cannot make use of wide FOV. Recently, ORB-SLAM3 [6] upgrades the previous system [2] and introduces the Kannala Brandt camera model [7] to handle fisheye cameras with a field of view over 180 degrees. Similarly, omnidirectional DSO [8] modifies DSO [4] for fisheye and catadioptric cameras by using the unified camera model [9] and achieves performance enhancement. However, these methods still cannot deal with 360 cameras.

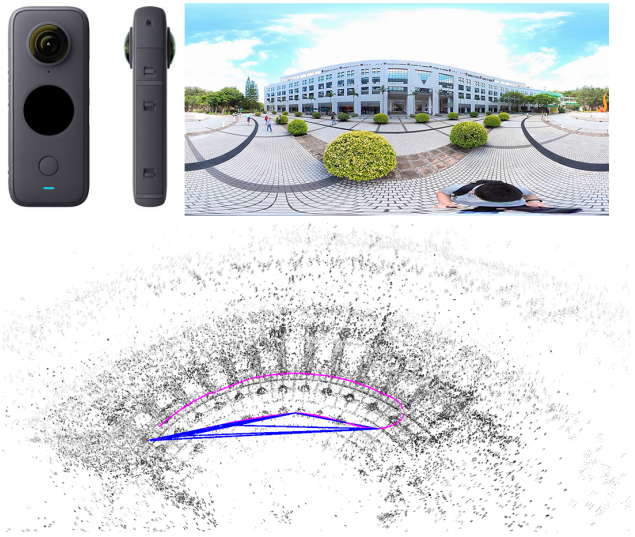


Fig. 1. Top left: A representative 360 camera that only consists of two ultra-wide FOV lenses. Top right: An example of its imaging. With the maturity of lens manufacture, ultra-wide FOV lenses have become cheaper and have high imaging quality. Bottom: the corresponding map reconstructed by our proposed system.

To take advantage of a single 360 camera, we propose a novel direct visual odometry method named 360VO. The system schematic is illustrated in Fig. 2. Similar to DSO, 360VO relies on optimizing photometric residuals instead of geometric keypoint correspondence to perform tracking. Specifically, 360VO exploits the spherical camera model to represent omnidirectional field of view. According to the specific model, we deduce epipolar constraints and adjust the mapping algorithm. It maintains a sliding window of activated keyframes and jointly optimizes their camera intrinsic parameters, pose estimations, affine brightness parameters and inverse depth of points. To validate the efficacy of the proposed method, on the one hand, we use a hand-held 360 camera to conduct experiments in indoor and outdoor environments. On the other hand, we utilize a simulator to render 10 sequences as the ground truth and then compare our system with another indirect method quantitatively. The results show our methods achieve state of the art performance. To the best of our knowledge, this is the first direct visual odometry system designed for the 360 camera.

Huajian Huang and Sai-Kit Yeung are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. hhuangbg@connect.ust.hk, saikit@ust.hk

This research project is partially supported by an internal grant from HKUST (R9429) and the Innovation and Technology Support Programme of the Innovation and Technology Fund (Ref: ITS/200/20FP).

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812203

II. RELATED WORKS

VO and VSLAM rely on the information extracted from images to estimate camera's poses and reconstruct a 3D representation of the world. Various systems (e.g., PTAM [10], SVO [1], LSD-SLAM [3], DSO [4], ORB-SLAM2 [5], etc.) have been proposed and can deal with monocular, stereo and RGBD cameras, significantly stimulating the development of our community. But it is still an acknowledged problem that visual systems are vulnerable in dynamic and texture-less environments, in particular for the monocular system.

To enhance the robustness of the system, Dual-SLAM [11] introduces a recovery mechanism to handle tracking loss. When the system fails to estimate pose of the current frame, it would initialize a new map to process incoming frames and then propagate the map backward in time to recover the map, which enables the system to return normal quickly without the necessity of revisiting the place. Additionally, explicit dynamic object detection and removal is an effective way to improve the accuracy of pose estimation. For example, DS-SLAM [12] and DynaSLAM [13] employ semantic segmentation networks and motion consistency constraints to detect and discard dynamic elements. DSLAM [14] utilizes point correlation to separate dynamic points avoiding impairment of pose estimate. EM-fusion [15] jointly infer dynamic objects camera poses by a probabilistic formulation which is capable of robust tracking and mapping in dynamic scenes.

Apart from the perspective of pure algorithm enhancement, another widely used solution is sensor fusion. The introduction of inertial measurement unit (IMU) [16]–[18] and depth sensors (e.g., Lidar) [19], [20] can provide redundancy and allow the system works in vision-weak environments. But it generally leads to complicated calibration and cost increases. In fact, the most straightforward solution is to widen the field of view.

Wide FOV cameras can provide images containing sufficient texture and theoretically allows for tracking visual landmarks over longer periods which are key ingredients to robust pose estimations [21]. Furthermore, it can maximize the proportion of static parts in the images such that the impact of dynamic elements is mitigated. And then implicit dynamic feature rejection methods, such as RANSAC and graphic optimization are competent to discard outliers and get accurate estimates in dynamic environments. Omnidirectional LSD-SLAM [22] and DSO [8] leverage the unified camera model [9] to extend LSD-SLAM [3] and DSO [4] for the omnidirectional camera respectively, while ORB-SLAM3 [6] starts to support fisheye camera input by incorporating the Kannala Brandt camera model [7]. In term of the multi-camera system, MULTICOL-SLAM [23] proposes a model that is applicable to arbitrary, rigidly coupled multi cameras. Similarly, ROVO [24] is an omnidirectional visual odometry method for a wide-baseline multi-camera system using a hybrid projection model. Four cameras are mounted to the rigid rig and have a 360 coverage of stereo observations of the environment. But their hardware setup and calibration procedure are complex. A more cost-efficient way to gain

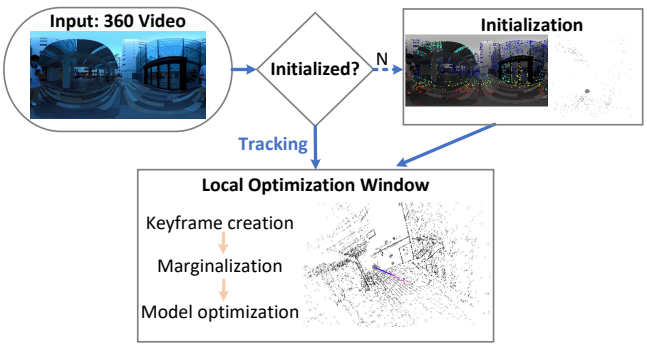


Fig. 2. System overview. The input of the system is an equirectangular frame sequence. After initialization, the system keeps tracking and optimizes relevant model parameters in the local window.

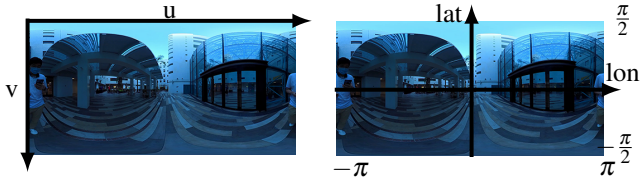
an omnidirectional perception is the 360 camera that consists of two back-to-back lenses with ultra wide FOV. Currently, existing systems that can handle 360 cameras are indirect methods that rely on geometric keypoints correspondences, e.g., OpenVSLAM [25]. Conversely, our proposed system 360VO takes advantage of photometric features, gradient of intensity, and seeks to estimate camera poses via optimizing photometric error. Moreover, 360VO leverages distinct epipolar constraints to restore depth of points. As it can make full use of the information of frames, the reconstructed model is denser than indirect methods' and exploration of unfamiliar environments is sped up. Owing to proper formulation, 360VO achieves robust tracking and low drift.

III. CAMERA MODEL

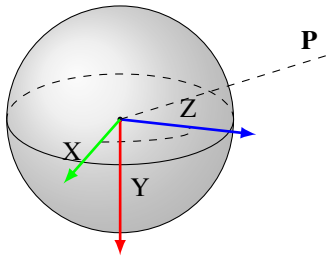
The camera model describes the mathematical relationship between the coordinates of a point in three-dimensional camera space Ω and its projection into the image space Ψ. Conventionally, image coordinates are denoted as $\mathbf{u} = [u,v]^T \in \Psi \subset \mathbb{R}^2$, 3D point coordinates are denoted as $\mathbf{X_c} = [X_c, Y_c, Z_c]^T \in \Omega \subset \mathbb{R}^3$. In addition, $\pi : \Omega \rightarrow \Psi$ represents the projection function which projects a 3D point in the camera space into a 2D pixel in the image space. The inverse process, up-projection function is π^{-1}. Generally, the pinhole camera model is used to describe perspective projection and the FOV it can represent is less than 180 degrees. When the FOV increases, features distortions increase non-linearly from the center to the side of the image. If we attempt to rectify a wide-angle image into a perspective projection, it would inevitably introduce interpolation artifacts. To keep high fidelity, the ideal representation of 360 camera is the spherical camera model and images is in equirectangular projection. The spherical camera model only needs two parameters $\mathbf{m} = [H,W]^T$, where H denotes image height and W denotes the width.

In 360VO, it involves the transformation among three coordinate systems, i.e., image space Ψ (Fig. 3(a)), spherical space Θ (Fig. 3(b)) and camera space Ω (Fig. 3(c)) coordinate systems. The projection function π is formulated as:

$$\pi(\mathbf{X_c}) = \begin{bmatrix} u \\ v \end{bmatrix} = K \begin{bmatrix} lon \\ lat \end{bmatrix} = K \begin{bmatrix} \arctan(X_c/Z_c) \\ -\arcsin(\hat{dY_c}) \end{bmatrix}, \quad (1)$$



(a) Image coordinate system (b) Spherical coordinate system



(c) Camera coordinate system

Fig. 3. Coordinate systems used in 360VO. It takes advantage of a spherical model to represent camera projection, and the 2D image is in equirectangular projection.

where, $\hat{d} = 1/\sqrt{X_c^2 + Y_c^2 + Z_c^2}$ is the inverse distance between the 3D point and center of unit sphere, *lon* and *lat* denote the longitude and latitude in spherical coordinate system respectively, $-\pi < lon < \pi$ and $-\pi/2 < lat < \pi/2$, while $\mathbf{K}$ is the camera intrinsic parameters and modeled as

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} = \begin{bmatrix} W/2\pi & 0 & W/2 \\ 0 & -H/\pi & H/2 \end{bmatrix}, \quad (2)$$

The function which up-projects image space coordinate system to camera space coordinate system is:

$$\pi^{-1}(\mathbf{u}, \hat{d}) = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \frac{1}{\hat{d}} \begin{bmatrix} \cos(lat)\sin(lon) \\ -\sin(lat) \\ \cos(lat)\cos(lon) \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} lon \\ lat \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x^{-1} & 0 & -f_x^{-1}c_x \\ 0 & f_y^{-1} & -f_y^{-1}c_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \quad (4)$$

Note that ideally we can infer the camera intrinsic parameters $\mathbf{K}$ as long as we determine the size (H, W) of the equirectangular image. In our system, $\mathbf{K}$ is optimized online.

IV. SYSTEM IMPLEMENTATION

In this section, we describe the pipeline of 360VO while the system overview is presented in Fig. 2.

A. Model Formulation

360VO is a direct VO method which estimates camera poses and the depth of features via minimizing photometric errors without a geometric prior. Following the definition of photometric error in DSO [4], the energy function of a pixel $\mathbf{p} \in \Psi$ in the host frame i regrading to a co-visible target frame j is

$$E_{\mathbf{p}}^{ij} = \sum_{\mathbf{u} \in N_{\mathbf{p}}} \|r\| = \sum_{\mathbf{u} \in N_{\mathbf{p}}} w_{\mathbf{u}} \left\| (I_j[\mathbf{u}'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[\mathbf{u}] - b_i) \right\|, \quad (5)$$

$$\mathbf{u}' = \pi(\mathbf{R}_{ji}\pi^{-1}(\mathbf{u}, \hat{d}^{\mathbf{p}}) + \mathbf{t}_{ji}),$$
$$\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ 0 & 1 \end{bmatrix} = \mathbf{T}_{ij} = \mathbf{T}_j \mathbf{T}_i^{-1}, \quad (6)$$

Here $I[\cdot]$ denotes the pixel intensity, a and b are affine photometric correction factors, and t is the exposure time. $N_{\mathbf{p}}$ represents a set of neighboring pixels included in the weighted sum of squared differences (SSD). In 360VO, it has 8 points, and each point shares the same inverse distance $\hat{d}^{\mathbf{p}}$. $w_{\mathbf{u}}$ is the gradient dependent weight, while $\|\cdot\|$ denotes the Huber norm. $\mathbf{T}_i^{-1}$ and $\mathbf{T}_j^{-1}$ are camera poses of the host and target frame respectively. $\mathbf{u}'$ is the corresponding point of $\mathbf{u}$ and reprojected by the relative camera pose from the host frame to the target frame, $\mathbf{T}_{ij}$. To reduce computational cost, the optimization of photometric residual is performed among the frames contained in the local window. Therefore, the complete energy function is :

$$\mathbf{E} = \sum_{i \in F} \sum_{\mathbf{p} \in P_i} \sum_{j \in obs(\mathbf{p})} E_{\mathbf{p}}^{ij}, \quad (7)$$

where F represents frames contained in local optimization window, P_i represents a set of selected points in the frame i and are randomly sampled from directional points with local gradients above a certain threshold, and $obs(\mathbf{p})$ represents the frames that can observe point $\mathbf{p}$.

Basically, to perform tracking of incoming frames which are considered as target frames, it attempts to find the poses $\mathbf{T}_j$ of the target frames minimizing energy function, $\arg\min_{T_j} \sum_{\mathbf{p} \in P_i} E^{ij}$. In term of local window optimization, it optimizes entire model parameters $\mathbf{M} = (\mathbf{T}_i, \mathbf{T}_j, \hat{d}, \mathbf{K}, a_i, b_i, a_j, b_j)$ using bundle adjustment. This process can be formulated as

$$\arg\min_{M} \mathbf{E}. \quad (8)$$

B. Initialization

To initialize tracking, we set the first frame as the initial frame and select the points in different layers of the feature pyramid based on the max gradient of pixel intensity. The inverse depth of candidate points is set as 1. For each following frame, it consistently optimizes the current camera pose, affine brightness parameters and inverse depth estimations from the lowest scale to the top scale. Once optimization has converged and sufficient frames pass by, the coarse initialization is successful and it starts to track incoming frames. Compared to the classical monocular system, 360VO makes use of omnidirectional perception and obtains the depth covering the agent around from two frames.

C. Epipolar Constraints

After the system successfully estimates camera poses, the mapping process is performed to incrementally reconstruct a 3D representation of the observed environment. Indirect methods benefit from the robustness of geometric features and can reject outliers according to feature descriptor matching. As a result, they can directly initialize the depth of valid points using triangulation. However, 360VO using

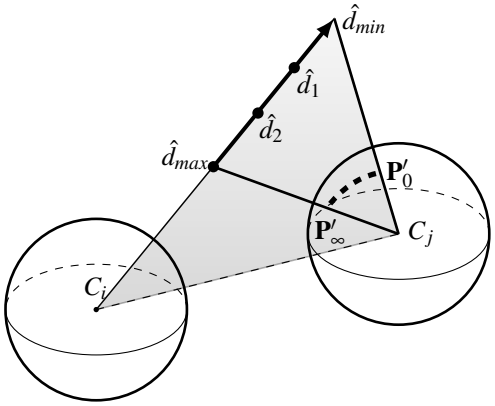


Fig. 4. Epipolar constraints. When tracking succeeds, it needs to create new activated points and refine their inverse depth via triangulation. High corresponding points of host frame c_i lie in the epipolar curve instead of line in the target frame c_j.

photometric features lacks such an advantage. Similar to other direct methods [1], [3], [4], we model the inverse depth estimation of a point with a probability distribution. Initially, it assumes the true value of depth lies in a large interval ($\hat{d}_{max}$, $\hat{d}_{min}$). With new tracked frames coming in, it continuously searches for the best corresponding points which minimize the photometric error Eq. (5). The best matches are used to refine distribution estimation. Once the depth search interval becomes small enough, the point is considered mature. And then the mature point is inserted into the map and optimized in the local optimization window. To speed up search and ensure precision, the searching of the corresponding point should obey epipolar constraints.

We let ρ to represent epipolar plane and S to represent the unit sphere. The epipolar constraints in camera space Ω coordinate system can be depicted as

$$\begin{cases} \rho: & aX+bY+cZ+d=0 \\ S: & X^2+Y^2+Z^2=1 \end{cases}, \tag{9}$$

And then epipolar constraints ε in spherical space is

$$a\cos(lat)\sin(lon)-b\sin(lat)+c\cos(lat)cos(lon)+d=0, \tag{10}$$

This implicit function represents a curve in image space. Therefore, different from the method using pinhole camera model, our system using spherical camera model needs to search potential points along the epipolar curves Eq. (10) instead of lines, as Fig. 4 shows. To perform iterative search along the epipolar curve, Omnidirectional DSO [8] leverages linear interpolation to approximate the curve. Supposed two points $\mathbf{P}'_0, \mathbf{P}'_\infty \in \Omega$ lie in the unit sphere C_j and correspond to the maximum and minimum inverse distance, so,

$$\mathbf{P}'_0 = \pi_s(\mathbf{R}_{ji}\pi^{-1}(\mathbf{p}, \hat{d}_{min}) + \mathbf{t}_{ji}), \tag{11}$$

$$\mathbf{P}'_\infty = \pi_s(\mathbf{R}_{ji}\pi^{-1}(\mathbf{p}, \hat{d}_{max}) + \mathbf{t}_{ji}), \tag{12}$$

here π_s denotes the function projecting the 3D points onto the unit sphere. The epipolar curve in image space is

$$\mathbf{u}(\alpha) = \pi(\alpha\mathbf{P}'_\mathbf{0} + (1-\alpha)\mathbf{P}'_\infty), \alpha \in [0,1], \tag{13}$$

Algorithm 1 Mapping

Input: Frame i with candidate points $\mathbf{P} \in \Psi$, reference frame j, and relative pose T_{ij}
Output: Activated points

```
for point p ∈ P do
    initialize search interval (d̂_max, d̂_min)
    P'_0 ← Eq. (11),   u'_0 = [u'_0, v'_0] ← π(P'_0)
    P'_∞ ← Eq. (12),   u'_∞ = [u'_∞, v'_∞] ← π(P'_∞)
    the normal of epipolar plane, n = [a,b,c]^T ← P'_0 × P'_∞
    du' = (u'_∞ − u'_0)/steps
    u' ← u'_0
    while i < steps do
        v' ← Eq. (15)
        E^ij_p ← Eq. (5)
        record the smallest value E_best and corresponding pixel [u', v']
        u' ← u' + du'
        i ← i + 1
    end while
    update search interval (d̂_max, d̂_min)
    if E_best < threshold then
        activate p
    end if
end for
```

The searching of the corresponding point starts at $\mathbf{u}(0)$ and re-computation of α is necessary for each interaction which involves first-order Taylor approximation. However, such a mathematical formulation requires cameras moving slowly. Otherwise, the errors will significantly fluctuate. This is because the initial uncertainty of depth is larger and search interval increases, linear interpolation cannot properly approximate a long curve. To further increase accuracy, we can add quantiles $\hat{d}_1, \hat{d}_2, ...$ to segment search interval, as Fig. 4 shows. As the number of quantiles increases, the incremental search with a constant step can be performed over each interval, which does not introduce heavy computation.

Although these methods are efficient, they are still approximations. In contrast, after solving the implicit function Eq. (10), 360VO conducts the corresponding point searching with a constant step along the horizontal direction ($u-axis$) of the image. We take the derivative of Eq. (10) with respect to lat and then we can get

$$\frac{\partial\varepsilon}{\partial lat} \rightarrow lat = \arctan\frac{-b}{a\sin lon + c\cos lon} + \Delta_{constant}, \tag{14}$$

where $\Delta_{constant} = lat_0 - \arctan\frac{-b}{a\sin lon_0 + c\cos lon_0}$ and $[lon_0, lat_0]^T$ is calculate with Eq. (1) and Eq. (11). Integrating Eq. (4), we can further get the explicit function of the epipolar curve in image space:

$$v = f(u) = \arctan\frac{-b}{a\sin\frac{(u-c_x)}{f_x} + c\cos\frac{(u-c_x)}{f_x}} + \Delta_{constant}. \tag{15}$$

Accordingly, we incrementally search for the corresponding point with a constant step along the $u-axis$ and compute

v using Eq. (15) for each step. The complete mapping algorithm of 360VO is depicted in the Algorithm 1.

D. Local Window Optimization

In addition to robustness, processing time is another important factor in practice. To make a trade-off between efficiency and accuracy, 360VO performs local optimization in the back end. The sliding optimization window managers 7 activated keyframes and 2500 map points. When the transformation from the reference keyframe to the latest tracked frame exceeds the threshold, a new keyframe will be created. Moreover, since 360VO relies on photometric features which are sensitive to illumination change, it is necessary to create a new keyframe when the relative brightness factor changes considerably. On the other hand, the keyframe which can observe enough activated points will be flagged as a marginalizing frame. Apart from utilizing feature correspondence, the system using a common camera model can determine invisible points according to transformation. The point that is no longer reprojected to the current frame would be inactive. However, theoretically, a 360 camera can capture omnidirectional information and the reference points can be reprojected onto current frames disregarding camera transformation. But, in general, the focal length of 360 camera is extremely short and the angular resolution of the image is low while the resolution is fixed. Therefore, if the distance between the reference and the latest keyframe is larger than the threshold, we marginalize the reference frame even though the local window is not filled.

As photometric error formulated in Eq. (8), we use the Gauss-Newton algorithm to jointly optimize all model parameters, including pose, camera intrinsic parameter and photometric factors. The Jacobin is defined as

$$\mathbf{J}_{\mathbf{M}=(\mathbf{T}_i,\mathbf{T}_j,\hat{d},\mathbf{K},a_i,b_i,a_j,b_j)} = \left[\frac{\partial r((\delta+x)\boxplus\zeta_0)}{\partial\delta}\right], \qquad (16)$$

where $\zeta_0 \in SE(3)$ and $\boxplus$ denotes the operation: $se(3) \times SE(3) \to SE(3)$.

V. EVALUATION

In this section, we extensively evaluate our proposed visual odometry, 360VO, on both synthetic and real-world scenarios. To conduct quantitative analysis, we propose a synthetic dataset with dense ground truth pose for each frame. Specifically, we employed the Unreal 4 engine to render 360 video sequences in the realistic urban scene models provided by [26]. The motion of the camera imitates characteristics of human, car and UAV in order to make the dataset more diverse and representative. In this way, the sequences cover distinct scenarios with different motion speeds and visual angles. The dataset consists of 10 sequences. On average each sequence has more than 2k frames, and the resolution of the frame is 1920×960. Representative frames of some sequences are demonstrated in Fig. 5. Besides the evaluation on the synthetic dataset, we also used a hand-held 360 camera to collect data in both indoor and outdoor environments in order to further verify the system's performance.

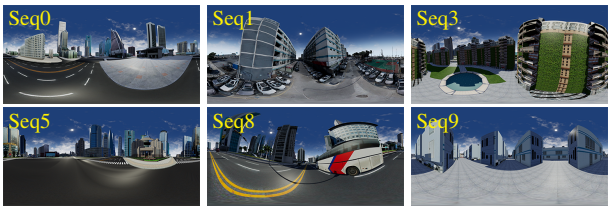


Fig. 5. Illustrations of parts of sequences in our synthetic dataset. The dataset is composed of 10 large-scale video sequences and rendered in realistic urban models.

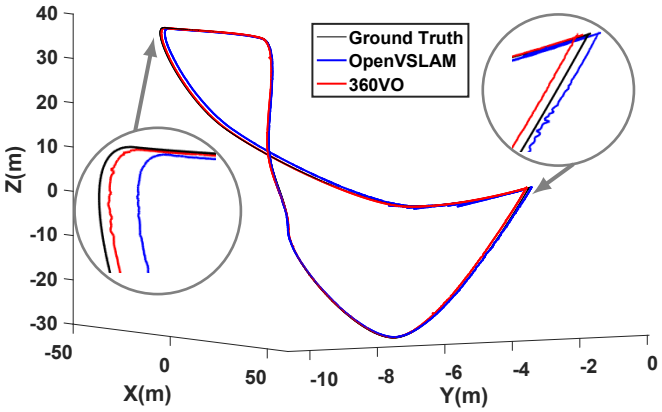
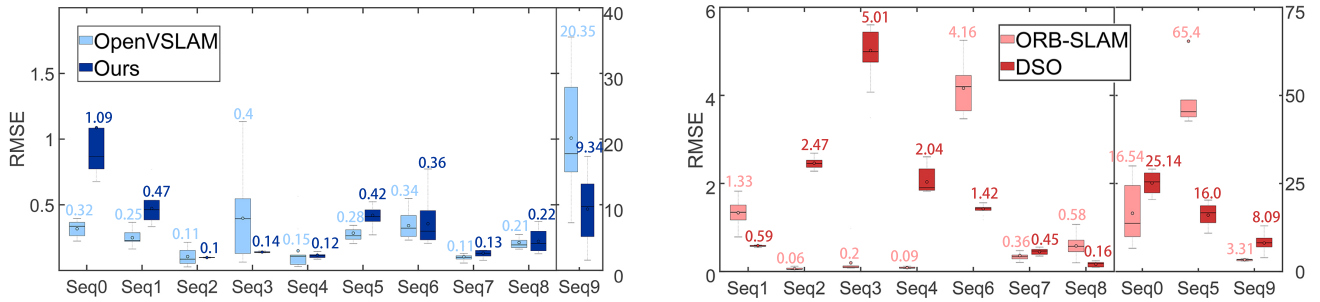


Fig. 6. Comparison of trajectory on sequence 3. The Ground-truth is in black, OpenVSLAM is in blue while ours 360VO is in red. The trajectory of 360VO is closer to the ground truth.

A. Quantitative Results

With the synthetic dataset, we compared our system with the indirect method, OpenVSLAM [25]. OpenVSLAM relies on sparse ORB features and supports various types of camera models, including the 360 camera. As systems are non-deterministic, we perform 10 trials on each sequence and then compute and report root mean square error (RMSE) between the ground truth and estimated trajectory, presented in Fig. 7 (left). Ours 360VO achieves competitive accuracy compared to OpenVSLAM. 360VO runs stably and has lower errors on sequences 2,3 and 4, while OpenVSLAM has better results on sequence 0 and the precision gap is noticeable. An comparison of trajectory is illustrated on Fig. 6 When the agent repeatedly revisits previous places, the system using global bundle adjustment can reduce the accumulated drift. The scene of sequence 9 is relatively monotonous and contains lots of similar buildings. Repetitive features affect the feature matching and increase errors of pose estimations. 360VO outperforms OpenVSLAM on sequence 9, but both systems have large errors.

Meanwhile, we also conducted an experiment to compare the systems using different camera models. We unwrap the equirectangular images to perspective images in order to form an affiliated dataset. The perspective images are of 640×640 resolution and have 90 degree FOV. And then we take them as input to run ORB-SLAM [2] and DSO [4]. As Fig. 7 shows, in general, wide FOV is able to improve the system's performance in terms of robustness and accuracy.



(a) Using 360 images (b) Using normal images

Fig. 7. Results on the synthesis dataset. Each sequence is run 10 times, and RMSE(m) of the trajectory is reported. The number at the top of each bar is the mean of RMSE. Ours 360VO achieves comparable results in contrast to OpenVSLAM. In addition, we rectify and crop the 360 images to perspective images of 90^o FOV, and take them as input to run ORB-SLAM and DSO. It is obvious that the methods utilizing 360 camera are commonly more robust and precise.

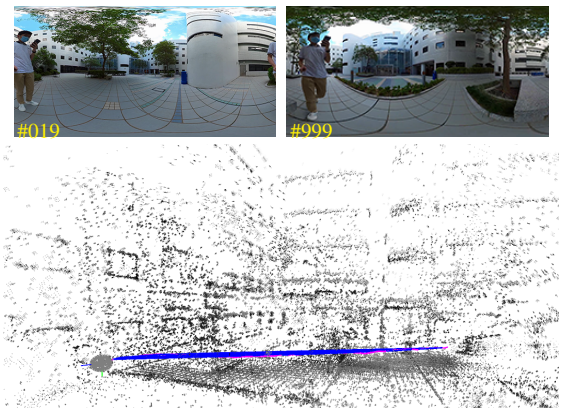


Fig. 8. Qualitative results in the outdoor environment.

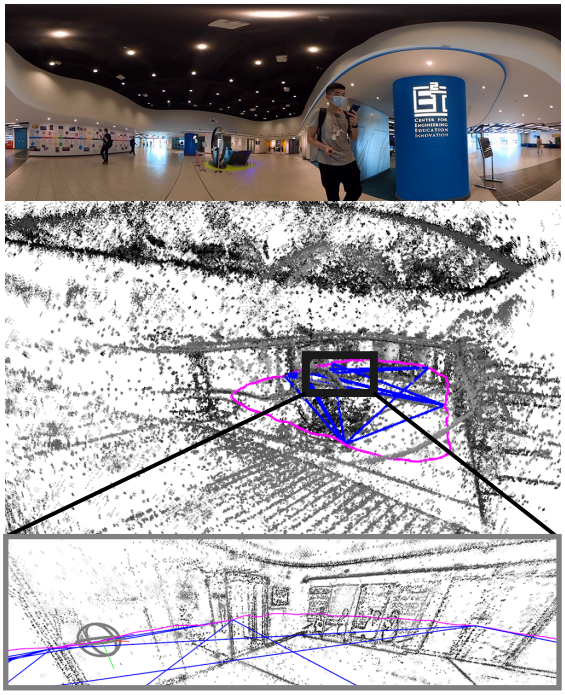
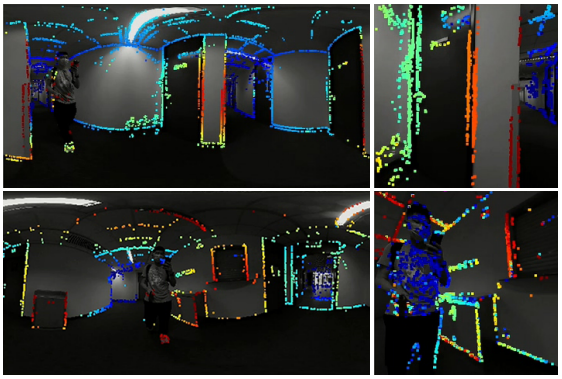


Fig. 9. Constraints between activated keyframes in the local optimization window are represented by blue lines, while Magenta curve denotes camera trajectory. The gray sphere denotes the current frame's position, while black points denote the 3D map. Since the same landmarks can be observed for a longer period, it has great consistency and low drift.



(a) 360VO (a) DSO

Fig. 10. 360 camera can capture sufficient features spatially even in the narrow indoor environment with textureless floors, white walls and dynamic objects. This inherent advantage allows 360VO to succeed in tracking and reconstruction, while the system using perspective images is prone to drift. Note: The color on the image represents the estimated depth of the point, near (red) → far (blue), best viewed in color.

B. Qualitative Results

The qualitative results illustrated in Fig. 8-10 are collected from real-world scenes. As it makes use of the information in the image, including edges and shades, 360VO is able to recover semi-dense point clouds of the observed environments. In addition, it is inevitable for 360 cameras to include the agents and manipulators. However, since these dynamic elements only occupy a small part of images, RANSAC and other optimization algorithms are qualified to remove them as outliers. Consequently, the influence regarding accuracy is limited, as Fig. 10 (bottom row) shows. Please refer to the attached video for more results.

VI. CONCLUSION

In this paper, we propose a novel direct visual odometry, referred to as 360VO, for a monocular 360 camera. It takes advantage of the spherical camera model to represent an omnidirectionally perception view and deduces distinct epipolar constraints. The system jointly optimizes camera intrinsic/extrinsic parameters, inverse distance, and photometric factors online. Comprehensive experiments on synthetic and real scenarios prove the effectiveness of 360VO.

REFERENCES

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

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[5] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[6] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[7] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[8] 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*, vol. 3, no. 4, pp. 3693–3700, 2018.

[9] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *European conference on computer vision*. Springer, 2000, pp. 445–461.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[11] H. Huang, W.-Y. Lin, S. Liu, D. Zhang, and S.-K. Yeung, "Dual-slam: A framework for robust single camera navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4942–4949.

[12] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[13] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[14] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "Rgb-d slam in dynamic environments using point correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[15] M. Strecke and J. Stuckler, "Em-fusion: Dynamic object-level slam with probabilistic data association," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5865–5874.

[16] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[17] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[18] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2510–2517.

[19] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1004–1011, 2021.

[20] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *arXiv preprint arXiv:2102.12400*, 2021.

[21] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 801–808.

[22] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct slam for omnidirectional cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 141–148.

[23] S. Urban and S. Hinz, "Multicol-slam-a modular real-time multi-camera slam system," *arXiv preprint arXiv:1610.07336*, 2016.

[24] H. Seok and J. Lim, "Rovo: Robust omnidirectional visual odometry for wide-baseline wide-fov camera systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6344–6350.

[25] S. Sumikura, M. Shibuya, and K. Sakurada, "Openvslam: a versatile visual slam framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2292–2295.

[26] Y. Liu, F. Xue, and H. Huang, "Urbanscene3d: A large scale urban scene dataset and simulator," 2021.