Edge Enhanced Direct Visual Odometry

Xin Wang[1]
xinwang_cis@pku.edu.cn
Dong Wei[1]
dongwei92@pku.edu.cn
Mingcai Zhou[2]
mingcai.zhou@samsung.com
Renju Li[1]
lirenju@3dscan.com.cn
Hongbin Zha[1]
zha@cis.pku.edu.cn

[1] Key Laboratory of Machine Perception
School of EECS
Peking University
Beijing, China

[2] Advanced Research Lab
Samsung Research Center-Beijing
Beijing, China

Abstract

We propose an RGB-D visual odometry method that both minimizes the photometric error and aligns the edges between frames. The combination of the direct photometric information and the edge features leads to higher tracking accuracy and allows the approach to deal with challenging texture-less scenes. In contrast to traditional line feature based methods, we use all edges rather than only line segments, avoiding aperture problem and the uncertainty of endpoints. Instead of explicitly matching edge features, we design a dense representation of edges to align them, bridging the direct methods and the feature-based methods in tracking. Image alignment and feature matching are performed in a general framework, where not only pixels but also salient visual landmarks are aligned. Evaluations on real-world benchmark datasets show that our method achieves competitive results in indoor scenes, especially in texture-less scenes where it outperforms the state-of-the-art algorithms.

1 Introduction

Visual odometry (VO)[17] focuses on estimating the camera motion from a sequence of images. VO more concerns the trajectory of a camera rather than a global map. Therefore, VO is considered as a subproblem of visual simultaneous localization and mapping (vSLAM). According to the type of inputs, VO can be mainly divided into monocular [4], stereo [9], and RGB-D [17] three types. VO is widely used in robotics and augmented reality (AR) [11]. Inevitably, there are texture-less scenes that VO may encounter [19]. These scenes illustrated in Fig. 4 are challenging, where many prevalent VO methods do not work. Taking an office as an example. Small robots (i.e. sweeping robots) lower than desks may be confused with the white wall and the floor without texture information; an AR helmet may lose track on the white ceiling. In view of this, it is crucial to strengthen the robustness of VO in texture-less scenes.

In this paper, we present an RGB-D VO approach where camera motion is estimated using the RGB images of two frames and the depth image of the first frame. Depth images captured by RGB-D camera only provide reliable pixel-wise depth measurements. Thus, tracking is indeed still monocular and is easy to expand to monocular version. Edges are natural and robust features prominent in most man-made environments, especially texture-less scenes. To overcome the difficulties in texture-less scenes we propose a semi-dense visual odometry method that not only utilizes pixel-wise information but includes entire edge features with a novel representation. The key contributions include:

- A robust registration method utilizing global edge features on an image instead of line segments, which no longer suffers from the aperture problem and the uncertainty of line segment endpoints.
- A dense representation for edge features which compresses the feature matching step and reprojected error minimizing step into one single stage, where optimization on image alignment and feature registration can be performed altogether.
- A general framework bridging the feature-based methods and the direct methods, which not only uses entire pixel information but also distinguishes valuable landmarks via aligning edges.

2 Related Work

A typical point feature based tracking approach first extracts designed features; then it restores geometric transformation between images with valid matches of these feature points [3][11][14]. These methods only use the information in a small region around the specific key points which rely heavily on texture, producing unsatisfying results on texture-less scenes. To deal with such a disadvantage, direct methods [15][5] minimize an energy function over all pixels on images and involve relatively more information. In practice, however, direct photometric information is less discriminative than visual features.

Besides point features, edge features are natural and informative which have been receiving great attention[16][12]. It is acknowledged that edge features are more robust to light variance, motion blur and occlusion than point features. Specifically, edges are more prominent than any other information in texture-less scenes. Previous researches mainly use line segments rather than comprehensively utilizing edges for the difficulties with describing and matching edges. Camillo et al. [2] extract line segments and minimize the distances between line segments represented by their endpoints. Hirose et al. [8] design a Line-based Eight-directional Histogram Feature (LEHF) for a line segment, making the edge feature descriptor more informative. Line segments only partially use the edge information and are prone to mistakes such as the aperture problem, which are caused by the homogeneity of lines and the uncertainty of the endpoints. Tarrio and Pedre [20] use all edges but search the correspondences in the normal directions, which is complicated.

There are researches focusing on fusing these methods and compensating for each other. Lu and Song [13] fuse point and line features in RGB-D VO to deal with lighting variations and uneven feature distributions. Forster et al. [7] design a semi-direct method that optimize the photometric error of small patches around FAST corners.

Texture-less scenes are challenging for visual tracking. Kerl et al. [10] minimize both photometric error and geometric error on RGB-D datasets. This method, however, is highly dependent on the reliable depth data thus is hard to expand to a monocular version. Ta et

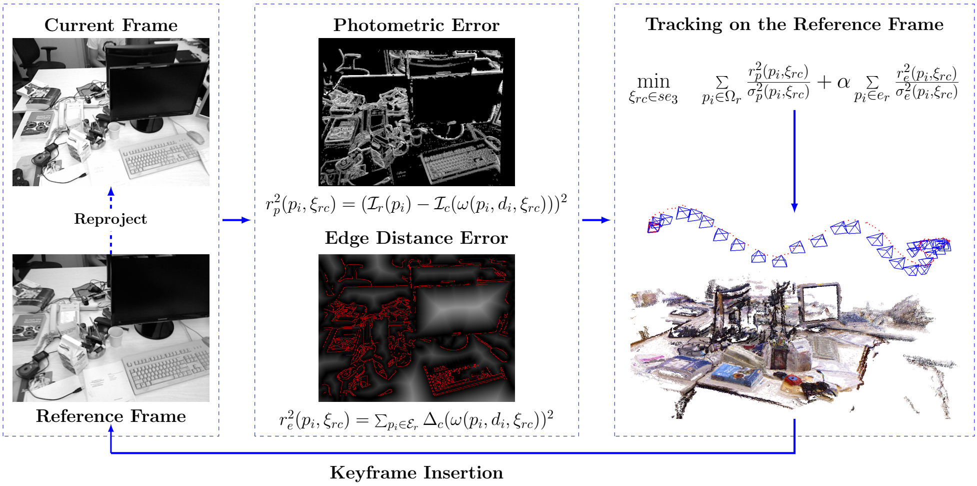

Current Frame
Reproject
Reference Frame
Photometric Error
$r_p^2(p_i, \xi_{rc}) = (\mathcal{I}_r(p_i) - \mathcal{I}_c(\omega(p_i, d_i, \xi_{rc})))^2$
Edge Distance Error
$r_e^2(p_i, \xi_{rc}) = \sum_{p_i \in \mathcal{E}_r} \Delta_c(\omega(p_i, d_i, \xi_{rc}))^2$
Tracking on the Reference Frame
$\min_{\xi_{rc} \in se_3} \sum_{p_i \in \Omega_r} \frac{r_p^2(p_i,\xi_{rc})}{\sigma_p^2(p_i,\xi_{rc})} + \alpha \sum_{p_i \in e_r} \frac{r_e^2(p_i,\xi_{rc})}{\sigma_e^2(p_i,\xi_{rc})}$
Keyframe Insertion

Figure 1: Overview of our approach

al. [19] propose a wall-floor intersection feature to deal with man-made texture-less indoor environments. This feature is mainly designed for the scenes that satisfy the Manhattan assumption such as a corridor, and is not tested on common desk-scale scenes referring to their paper.

3 Edge Enhanced Direct Visual Odometry

3.1 System Overview

As illustrated in Fig. 1, to estimate the trajectory of the camera from an RGB-D sequence incrementally, every new frame (current frame $\mathcal{F}_c$) is aligned to a reference frame $\mathcal{F}_r$, which is a carefully selected keyframe defined in Section 3.4.5. First, the visual edges are extracted in $\mathcal{F}_c$ (edges in $\mathcal{F}_r$ have been extracted beforehand). Then, error caused by camera pose at $\mathcal{F}_c$ is estimated: non-edge points in $\mathcal{F}_r$ are reprojected to $\mathcal{F}_c$, followed by the computation of photometric error defined in Section 3.4.1; meanwhile, edge points in $\mathcal{F}_r$ are reprojected to a distance field (Section 3.3) derived from edges in $\mathcal{F}_c$. Finally, motion is recovered in an optimization schema: the edge distance error (Section 3.4.2) is minimized along with the photometric error. To deal with the fast accumulation of drifting error, a key frame strategy is used.

3.2 Geometric Notations

RGB-D visual odometry is aimed to estimate the camera trajectory from an RGB-D stream. A typical RGB-D camera such as Microsoft Kinect generates a pair of RGB image $\mathcal{I}_t$ and depth image $\mathcal{D}_t$ at the timestamp t. The RGB image and the depth image are registered and are correspondent at pixel level. For an RGB image pixel $p = (p_x, p_y)^T$, $\mathcal{I}_t(p)$ denotes the intensity and $\mathcal{D}_t(p)$ the depth at p.

For a 3D point, $X = (x,y,z)^T$ denotes its position. The intrinsic parameters of a camera used in our model consist of the focal length f_x, f_y and the camera center c_x, c_y. We project 3D points from the camera coordinate system to the image plane by

$$p = \pi(X) = (\frac{xf_x}{z} + c_x, \frac{yf_y}{z} + c_y)^T, \tag{1}$$

where $\pi(X)$ is the projection function. As for its inverse, we use $\pi^{-1}(p,d)$ to represent the transformation from a pixel coordinate p and its correspondent depth d to a 3D point

$$X = \pi^{-1}(p,d) = (\frac{p_x - c_x}{f_x}d, \frac{p_y - c_y}{f_y}d, d)^T. \tag{2}$$

The camera motion is a rigid transformation represented in Lie group:

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \in SE(3),\ R \in SO(3),\ t \in \mathbb{R}^3, \tag{3}$$

where R is the rotation matrix and t is the translation vector. Specifically, we use T_{wc} to denote the transformation from the world coordinate system to the camera coordinate system:

$$X_c = T_{wc} X_w, \tag{4}$$

where X_c and X_w denote the coordinate of a 3D point in the camera and the world coordinate system.

T is an over-parameterized representation. We use a six dimensional vector $\xi \in se(3)$ as a compact representation, which is correspondent to $T \in SE(3)$. Exponential map $T = \exp(\xi)$ in Lie algebra does the conversion.

Given a transformation $T_{ij} = \exp(\xi_{ij})$ from the frame i to j, a pixel p of frame i with the depth $\mathcal{D}_i(p)$ can be reprojected to the pixel $\tilde{p}$ of frame j by the warp function

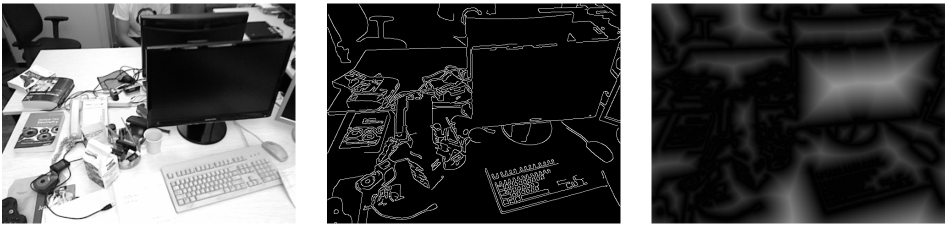
$$\tilde{p} = \omega(p,d,\xi_{ij}) = \pi(T_{ij}\pi^{-1}(p,\mathcal{D}_i(p))). \tag{5}$$

3.3 Edge Distance Transform

We apply the Canny edge detector[1] on each RGB image to extract edges. An example result is showed in Fig. 2. Edges in an image is marked in a set $\mathcal{E} = \{p_i \mid l(p_i) = 1\}$, where a mask function $l(p)$ denotes whether a pixel is an edge point

$$l(p) = \begin{cases} 1, & p \text{ is an edge point} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

On the assumption that a correct motion between the reference frame $\mathcal{F}_r$ and the current frame $\mathcal{F}_c$ is recovered, the reprojected edges from $\mathcal{F}_r$ to $\mathcal{F}_c$ coincide with the edges observed in $\mathcal{F}_c$. To find the correct motion between $\mathcal{F}_r$ and $\mathcal{F}_c$, the edges should be aligned. As edge features are sparse and hard to describe (in the aspect of well-defined descriptors), we give edges a dense representation using the distance transform algorithm in order to align edges between frames.



(a) Image (b) Edge (c) Distance transform

Figure 2: (a) is a frame image. (b) is the result of Canny edge detection from (a), where the edges are labeled by white pixels. (c) is a distance transform image derived from edges in (b), where the intensities reflect the distance field: whiter regions are further to edges.

Distance transform is a representation of an image, usually appears in the form of distance field. We define a distance field $\Delta(p)$ holding the minimal distance to the nearest edge point for every pixel:

$$\Delta(p_i) = \min_{q \in \mathcal{E}} d(p_i, q), \tag{7}$$

where $\mathcal{E}$ is the set of edge points mentioned above. Fig. 2 illustrates a distance transform image. The intensity reflects the value of distance field: whiter regions are further to edges. Distance transform is a well researched problem. Many linear complexity algorithms such as [6] have been proposed. Depending on the extracted edges in $\mathcal{F}_c$, we can compute $\Delta(p)$ very fast.

3.4 Image Alignment

3.4.1 Photometric Error

Based on $\mathcal{D}_r(p)$ of $\mathcal{F}_r$ and the relative transformation ξ_{rc} from $\mathcal{F}_r$ to $\mathcal{F}_c$ using the warp function ω in Equation (5), the pixels in $\mathcal{F}_r$ are reprojected to $\mathcal{F}_c$, as illustrated in Fig. 3. The photometric error for pixel p_i is defined as

$$r_p^2(p_i, \xi_{rc}) = (\mathcal{I}_r(p_i) - \mathcal{I}_c(\omega(p_i, d_i, \xi_{rc})))^2, \tag{8}$$

where $\mathcal{I}_r(p_i)$ is the intensity of p_i in $\mathcal{F}_r$, and $\mathcal{I}_c(\omega(p_i, d_i, \xi_{rc}))$ in $\mathcal{F}_c$ after warping. These two values are expected to be consistent when the pose estimation ξ_{rc} is accurate.

In practice, smooth regions of an image are less useful for image alignment. Instead of all pixels, we use pixels whose intensity gradients are above a threshold, forming a subset Ω_r of the pixels in $\mathcal{F}_r$. The photometric error between $\mathcal{F}_r$ and $\mathcal{F}_c$ is then defined as

$$E_p(\xi_{rc}) = \sum_{p_i \in \Omega_r} r_p^2(p_i, \xi_{rc}) = \sum_{p_i \in \Omega_r} (\mathcal{I}(p_i) - \mathcal{I}(\omega(p_i, d_i, \xi_{rc})))^2. \tag{9}$$

3.4.2 Edge Distance Error

Similarly, we reproject edge points from $\mathcal{F}_r$ to $\mathcal{F}_c$ with the warp function ω. We use $\mathcal{E}_r$ and $\mathcal{E}_c$ to distinguish edge points in $\mathcal{F}_r$ and $\mathcal{F}_c$. Assuming a stable performance of Canny

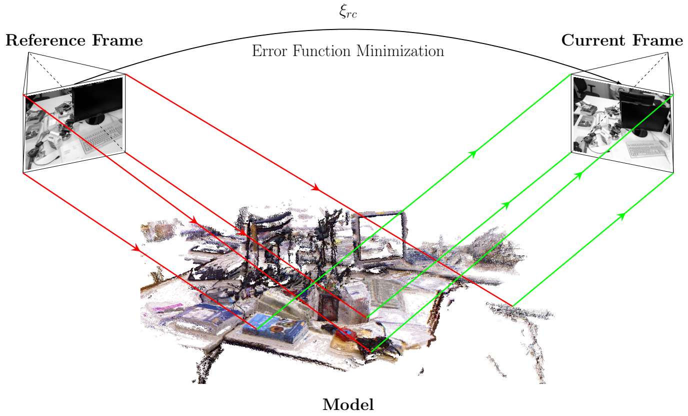


Figure 3: Computation of the error function. The pixels with depth estimation on the reference frame are reprojected to the current frame. The optimal ξ_{rc} is estimated by minimizing the error function.

detector on consecutive frames, $\mathcal{E}_r$ and $\mathcal{E}_c$ converge given a reasonable relative pose estimation. We compute the distance between $\mathcal{E}_r$ and $\mathcal{E}_c$ by summing up the values of $\Delta_c(p_i)$ after reprojecting every $p_i \in \mathcal{E}_r$ onto Δ_c

$$E_e(\xi_{rc}) = \sum_{p_i \in \mathcal{E}_r} r_e^2(p_i, \xi_{rc}) = \sum_{p_i \in \mathcal{E}_r} \Delta_c(\omega(p_i, d_i, \xi_{rc}))^2, \tag{10}$$

where Δ_c denotes the distance field computed with the edges in $\mathcal{F}_c$.

3.4.3 Error Function and Optimization

We organize our algorithm in an optimization fashion. Starting from the existing reference frame $\mathcal{F}_r$, we minimize the photometric error r_p^2 and the edge distance error r_e^2 altogether to get an optimal relative pose ξ_{rc}. By multiplying a weight α, we combine these two types of error and formulate an energy function

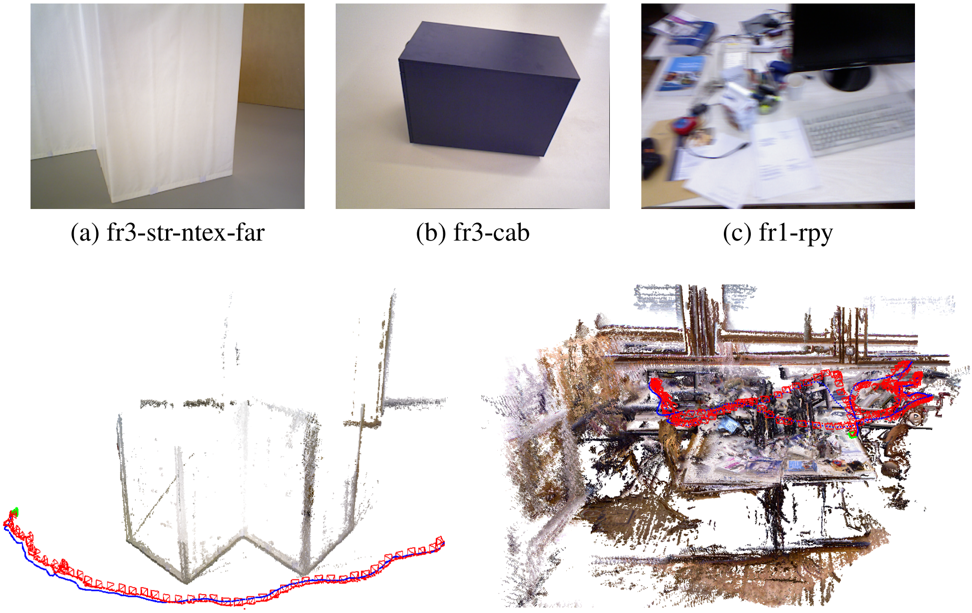
$$E(\xi_{rc}) = \sum_{p_i \in \Omega_r} \frac{r_p^2(p_i, \xi_{rc})}{\sigma_p^2(p_i, \xi_{rc})} + \alpha \sum_{p_i \in \mathcal{E}_r} \frac{r_e^2(p_i, \xi_{rc})}{\sigma_e^2(p_i, \xi_{rc})}, \tag{11}$$

where $\sigma_k(p_i, \xi_{rc})$, $k \in \{p, e\}$ are normalizing terms referring to [5] involving the gradient of residual on the inverse depth map:

$$\sigma_k^2(p_i, \xi_{rc}) = (\frac{\partial r_k(p_i, \xi_{rc})}{\partial \mathcal{D}_r^{-1}(p_i)})^2 V, \tag{12}$$

where $\mathcal{D}^{-1}(p)$ denotes the inverse depth at p, and V denotes an constant inverse depth variance. These weight terms are inspired by [5] where more details are discussed.

We apply Levenberg-Marquardt algorithm to minimize the proposed non-convex objective function with the initial value set to the relative pose between the last frame and $\mathcal{F}_r$. A



(a) fr3-str-ntex-far (b) fr3-cab (c) fr1-rpy

(d) fr3-str-ntex-far (e) fr1-desk2

Figure 4: Example scenes in the benchmark dataset. (a) and (b) are texture-less scenes. (c) is a desk scene with rich texture and heavy motion blur. (d) and (e) are the sample trajectories of the camera.

coarse-to-fine pyramid approach starts from low resolution is adopted to increase the convergence radius.

3.4.4 Edge Point Selection

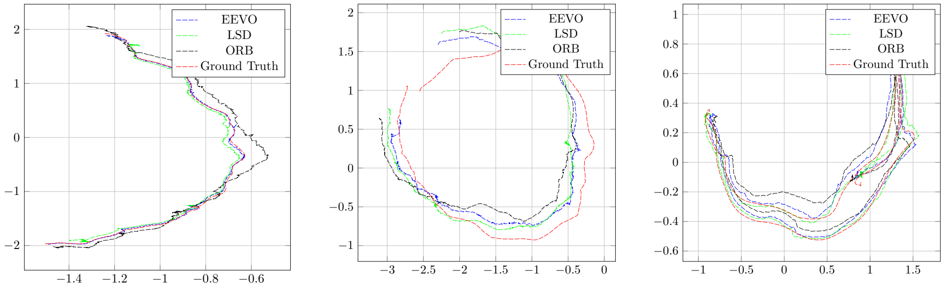
Inconsistent edge points are eliminated from the edge point set $\mathcal{E}_r$ concerning the robustness of the algorithm. The inconsistency comes from two major reasons: on the one hand, consecutive frames do not always hold the same edges due to fast motion or image blur; on the other hand, it might occur that some reprojected edges from the reference frame are in reality occluded in the current frame. We maintain a mean edge distance error $\bar{r}_e$ for each frame. For the current frame, edge points whose edge distances are β times large than $\bar{r}_e$ of the previous frame will be eliminated from $\mathcal{E}_r$. This process is performed on the image pyramid in order.

3.4.5 Keyframe Selection

We use a frame-to-keyframe method to avoid the fast accumulation of drift error. Current frame is tracked on the latest keyframe acting as the reference frame. A new keyframe is created and replaces the reference frame under several conditions: the distance between camera principles, or the angle between optical axises, or the edge distance error between two frames is larger than thresholds, which lead to small overlap between two frames; a large

sequence	EE-VO	LSD(VO)	ORB(VO)
fr3-str-ntex-far	**0.019**	0.067	0.110
fr3-str-ntex-far(v)	**0.018**	0.148	0.060
fr3-cab	**0.268**	0.322	0.384
fr1-desk	0.051	**0.044**	0.070
fr1-desk2	**0.067**	0.095	0.099
fr1-xyz	0.049	0.041	**0.008**
fr1-rpy	0.065	**0.064**	0.080

Table 1: RMSE of absolute trajectory error (ATE) in meters on TUM-RGBD dataset. The sequences with $fr3$ prefix capture texture-less scenes. $fr1$-$desk2$ and $fr2$-rpy suffer from heavy motion blur. LSD and ORB are all running in pure localization mode where global optimization is disabled.



(a)fr3-str-ntex-far (b) fr3-cab (c) fr1-desk2

Figure 5: Qualitative comparison of trajectories.

difference of edge point number between two frames due to fast motions or occlusions.

4 Experiments

Our system, named Edge Enhanced Direct Visual Odometry (EEVO), is implemented based on the open source software of [5]. The pyramid approach used in our system starts from the resolution of 80×60 and ended with 320×240 with step 2. Empirically, the weight α used to combine the photometric error and the edge distance error in Equation (11) is set in the range of $5^2 \sim 10^2$, and the threshold β discussed in 3.4.4 is set in the range of $2 \sim 6$. Generally, texture-rich scenes prefer lager α and β while texture-less scenes the opposite. To ensure the best performance, we manually adjust the parameters on some sequences.

We test our approach on the TUM RGB-D benchmark[18] on a PC with Intel Core i5-4590 CPU and 8GB memory, running at the speed of $27 \sim 39$ fps according to the content of the images. This benchmark contains multiple real world RGB-D sequences and provides accurate ground-truth trajectories. It has been widely used for evaluation for RGB-D visual odometry or RGB-D SLAM.

The sequences in Table 1 with $fr3$ prefix capture texture-less scenes illustrated in Fig. 4. $fr3$-str-$ntex$-far and $fr3$-str-$ntex$-$far(v)$ capture the same scene that only contains a white wall, as shown in Fig. 4 (a) and (d). Red dots in Fig. 4 (d) plot the trajectory recovered by our method, while blue dots demonstrate the ground truth trajectory. $fr3$-cab captures a smooth black cabinet, as shown in Fig. 4 (b). Sequences with $fr1$ prefix capture a desk-scale scene in an office with rich texture. The camera sweeps over four desks, as shown in Fig. 4 (e). The fast motion of camera generates heavy motion blur causing great difficulties in tracking, which is shown in Fig. 4 (c).

We evaluate our method both on texture-less scenes and common indoor scenes, and compare the result to the state-of-the-art methods for both the point feature based method ORB-SLAM and the direct method LSD-SLAM. It is worth it to mention that these two methods run in pure localization mode using RGB-D images without global optimization, since we define the problem in the aspect of visual odometry. The original implementation of ORB-SLAM refuses to track the sequences on texture-less scenes where feature points are insufficient. We modify ORB-SLAM and adjust some thresholds in ORB-SLAM and LSD-SLAM to enable tracking in some extreme conditions.

In our evaluations, a recovered trajectory is denoted with $\xi_1, \cdots, \xi_n \in se(3)$. Relatively, the ground truth trajectory is denoted with $\eta_1, \ldots, \eta_n \in se(3)$. The absolute trajectory error (ATE) referring to [18] is utilized to show the global consistency of the estimated trajectory. This error is computed after aligning the estimated trajectory to the ground truth with a rigid transformation S

$$E_i = Q_i^{-1} S P_i. \tag{13}$$

Table 1 lists the results of RMSE of ATE. Fig. 5 shows the trajectories on these datasets. The first three sequences in the table are captured in texture-less scenes. The results show that our method outperforms others on these three texture-less sequences. It is reasonable since edge features are prominent in texture-less scenes: the alignment of edges does improve the robustness. Our method also achieves challenging results on sequences with heavy motion blur, indicating its robustness to fast motion. As far as we concern, the heavy motion blur not only increases the difficulties of extracting point features, but also reduces the discrimination of intensities used in direct methods; edge features are more robust with our dense representation.

5 Conclusions and Future Work

In this paper, we present a real-time RGB-D visual odometry approach that fuses both the photometric information and the edge features, bridging the prevalent direct and feature-based methods. This combination is realized in a nature way by designing an elegant dense representation for sparse edge features. Through our experiments, we demonstrate that our method produces challenging results on common indoor scenes. Moreover, the fusion of two kinds of information significantly improve the robustness of tracking in texture-less scenes, comparing to the methods using either of them. In the future, we plan to expand our method to a monocular SLAM framework that fully use RGB images. Under the epipolar constraint, the edges in our framework provide spacial prior knowledge for stereo matching, which is able to reconstruct a more reliable semi-dense map.

Acknowledgements

This work was supported in part by the Beijing Municipal Natural Science Foundation (4152006).

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on PAMI*, 8 (6):679–698, 1986.

[2] E. Eade and T. Drummond. Edge landmarks in monocular SLAM. *Image and Vision Computing*, 27(5):588–596, 2009.

[3] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Proceedings of ICRA*, pages 1691–1696, 2012.

[4] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of ICCV*, pages 1449–1456, 2013.

[5] J. Engel, T. Schops, and D. Cremers. LSD-SLAM: Large-scale direct monocular S-LAM. In *Proceedings of ECCV*, pages 834–849, 2014.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Proceedings of ICRA*, pages 15–22, 2014.

[8] K. Hirose and H. Saito. Fast line description for line-based SLAM. In *Proceedings of BMVC*, 2012.

[9] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Proceedings of IROS*, pages 3946–3952, 2008.

[10] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for RGB-D cameras. In *Proceedings of IROS*, pages 2100–2106, 2013.

[11] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of ISMAR*, 2007.

[12] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *Proceedings of ECCV*, pages 802–815, 2008.

[13] Y. Lu and D. Song. Robust RGB-D odometry using point and line features. In *Proceedings of ICCV*, pages 3934–3942, 2015.

[14] J. R. Mur-artal and Montiel J. M. M. ORB-SLAM: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[15] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of ICCV*, pages 2320–2327, 2011.

[16] P. Smith, I. D. Reid, and A. J. Davison. Real-time monocular SLAM with straight lines. In *Proceedings of BMVC*, pages 17–26, 2006.

[17] F. Steinbrucker, J. Sturm, and D. Cremers. Real-time visual odometry from dense RGB-D images. In *Proceedings of ICCV*, pages 719–722, 2011.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of IROS*, pages 573–580, 2012.

[19] D. Ta, K. Ok, and F. Dellaert. Vistas and parallel tracking and mapping with wall-floor features: Enabling autonomous flight in man-made environments. *Robotics and Autonomous Systems*, 62(11):1657–1667, 2014.

[20] J. J. Tarrio and S. Pedre. Realtime edge-based visual odometry for a monocular camera. In *Proceedings of ICCV*, pages 702–710, 2015.