PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Light field SLAM based on ray-space projection model

Yaning Li, Qi Zhang, Xue Wang, Qing Wang

Yaning Li, Qi Zhang, Xue Wang, Qing Wang, "Light field SLAM based on ray-space projection model," Proc. SPIE 11187, Optoelectronic Imaging and Multimedia Technology VI, 1118706 (18 November 2019); doi: 10.1117/12.2538016

SPIE.

Event: SPIE/COS Photonics Asia, 2019, Hangzhou, China

Light Field SLAM based on Ray-Space Projection Model

Yaning Li, Qi Zhang, Xue Wang, and Qing Wang

School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, P.R. China

ABSTRACT

Pose estimation is the key step of simultaneous localization and mapping (SLAM). The relationship between the rays captured by multiple light field cameras can provide more constraints for pose estimation. In this paper, we propose a novel light field SLAM (LF-SLAM) based on ray-space projection model, including visual odometry, optimization, loop closing and mapping. Unlike traditional SLAM, which estimates pose based on point-point correspondence, we firstly utilize ray-space features to initialize camera motion based on light field fundamental matrix. In addition, a ray-ray cost function is presented to optimize camera pose and 3D points. Finally, we exhibit the motion map and 3D reconstruction results from a moving light field camera. Experimental results have verified the effectiveness and robustness of the proposed method.

Keywords: SLAM, light field, pose estimation, ray-space projection

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) plays a key role in the field of automatic driving, aviation, spaceflight and navigation. SLAM has been developed a systematic statement since it was first proposed by A.J.Davison as a real-time monocular visual SLAM system (MonoSLAM).[1] Afterwards, inspired from monocular SLAM system, some pratical SLAM systems were also developed successively, such as Parallel Tracking and Mapping (PTAM),[2] DT-SLAM,[3] LSD-SLAM,[4] SVO,[5] ORB-SLAM,[6] DPPTAM,[7] REBVO,[8] Direct Sparse Odometry.[9] At the same time, dual-purpose slam is also proposed and typical binocular slam has: LIBVISO2,[10] ORB-SLAM2,[11] S-PTAM,[12] ORBSLAM-DWO,[13] PL-StVO,[14] ScaViSLAM.[15] All the above methods estimate the pose just by feature points thus the obtained matching relationship is sparse and not robust.

Light field can solve the sparse and robust problem in the process of pose estimation. Light field cameras[16] can record spatial and angular information of light rays in 3D space. Levoy et al.[17]proposed a two-parallel Planes model to model 4D light field data $L(s,t,x,y)$, the parameters (s,t) and (x,y) represent the orientation and position information, respectively. Liang et al.[18] modeled the propagation process of light in Euclidean space, and then proposed a parameterization method equivalent to two-parallel plane model. Existing light field camera models[19,20]mostly define the projection from an arbitrary point in 3D space (passing through micro-lens) to corresponding pixel on the sensor. Nevertheless, light field essentially represents the collection of rays in space. In order to explore the relationship between rays in the process of pose estimation, ray-space projection model[21] which provides more constraints for pose estimation is used.

In this paper, we exploit a novel light field SLAM (LF-SLAM) based on ray-space projection model, including visual odometry, optimization, loop closing and mapping. Based on the rays' relationship, we first initialize the poses and world coordinates of each frame by ray-projection model and fundamental matrix on the front-end so that we can get a global feature $\boldsymbol{F}$. Secondly, in order to optimize the poses and 3D points, we propose a ray-ray cost function to minimize the distance among rays. Furthermore, loop closing is used to eliminate accumulative error. Subsequently, an effective mapping and 3D reconstruction is given. Finally, the quantitative and qualitative comparisons demonstrate the effectiveness and robustness of the proposed light field SLAM (LF-SLAM).

Our main contributions are

1) The ray-space projection model and fundamental matrix is used to initialize camera pose and 3D points.

2) On the back-end, a ray-ray cost function is proposed to optimize the pose and 3D points.

Further author information: (Send correspondence to Qing Wang)

Qing Wang: E-mail: qwang@nwpu.edu.cn , Telephone:+86 29 8843 1518

The work was supported by NSFC under Grants No.61531014 and No.61801396.

Optoelectronic Imaging and Multimedia Technology VI, edited by Qionghai Dai, Tsutomu Shimura, Zhenrong Zheng, Proc. of SPIE Vol. 11187, 1118706 · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2538016

2. RELATED WORK

SLAM model Generally speaking, SLAM can divide into two kind: monocular vision SLAM system and binocular SLAM system. MonoSLAM[1] is a typical monocular vision SLAM system which is proposed by A. J. Davison. MonoSLAM extends the kalman filtering as a back-end to track the sparse feature points on front-end. This system is the first time to realize on-line operation, nevertheless, there still exists some problem such as: narrow scene, limited landmarks and easily lost sparse feature. After that, J. Engel[4] proposed Large-Scale Direct Monocular SLAM, instead of using keypoints, it directly operates on image intensities both for tracking and mapping. LSD-SLAM[4] could runs in real-time on a CPU, and even on a modern smartphone. More recently, Mur-Artal *et al.* exhibit an ORB-SLAM.[6] This system is based on ORB (Orinted FAST and BRIEF) feature which have high efficiency compared to other features and it just need 33ms to calculate one frame of image. The latest monocular SLAM system has REBVO[8] and Direct Sparse Odometry.[9] REBVO tracks a camera in Real-time using edges, an on-board part doing all the processing and sending data over UDP and an OpenGL visualizer. Direct Sparse Odometry integrates a full photometric calibration, accounting for exposure time, lens vignetting, and non-linear response functions. As for binocular SLAM system, LIBVISO2[10]is an 8-point algorithm for fundamental matrix estimation and the majorization is based on minimizing the reprojection error of sparse feature matches. ORBSLAM-DWO[13] is developed on top of ORB-SLAM with double window, it supports monocular, stereo, and stereo + inertial input for SLAM.
Light field camera model Ng proposes a hand-held light field camera[22] in 2006, after that, some models for reconstructing 3D points[21]and projection model based on light field camera are proposed[20].[21] Bok *et al.*[19] proposes a 6-parameter geometric projection model for light field camera to estimate the intrinsic parameters. This model acquires line features from micro-lens images of raw data directly. Guo *et al.*[23]presents a ray-space motion matrix that describes how light field ray parametrization are transformed under different light field coordinates, however, this model only considers the relationship between point sets under Plücker coordinates. More recently, Zhang *et al.*[24] simplifies the light field camera geometry as a 4-parameter model and calibrate its intrinsic parameters, however this model can not fully explain light field camera geometry. Therefore, Zhang *et al.*[20] exhibits a 6-parameter multi-projection-center (MPC) model for light field cameras and this model also apple to traditional cameras. On the basis of MPC, Zhang[21] proposes a ray-space projection model and fundamental matrix among multiple light field cameras. this model can describe ray-ray relationship between light field cameras. In the work, we exploit the relationship between the rays captured by multiple light field to initialize the pose and world coordinates of each frame and a ray-ray cost function to optimize them.

3. LIGHT FIELD SLAM

3.1 Initialization of Camera Motion by Ray-Space Projection Model

Light field describes the relationship between the rays. In our LF-SLAM, utilizes this relationship we can initializes more exact positions and 3D points. Without loss of generality, we simplify our analysis between two 4D Light fields (L_1, L_2) to calculate fundamental matrix based on the rays.

Firstly, we parameterize the 4D light field in a *relatives* two-parallel plane coordinates.[17] There is a view plane with parameter (s-t) where $Z = 0$, and image plane ($Z = f$) with parameter (x-y). We normalize the spacing of the two-parallel plane to 1. In this parametrization, the rays in light field can be described by $\boldsymbol{r} = (s, t, x, y)^\top$. Then we can construct a mapping between spatial points and image plane coordinates (x, y),[21] λ=Z is the scale factor, as shown in Fig. 1.

Secondly, The ray of two light fields $L_1(i, j, u, v)$, $L_2(i, j, u, v)$ can be obtained by feature extraction, respectively. Then the light field $L(i, j, u, v)$ is transformed into a normalized undistorted physical light field $L(s, t, x, y)$ through an intrinsic matrix,[20] as shown in Eq. 1.

$$\begin{bmatrix} s \\ t \\ x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} k_i & 0 & 0 & 0 & 0 \\ 0 & k_j & 0 & 0 & 0 \\ 0 & 0 & k_u & 0 & u_0 \\ 0 & 0 & 0 & k_v & v_0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ u \\ v \\ 1 \end{bmatrix} \tag{1}$$

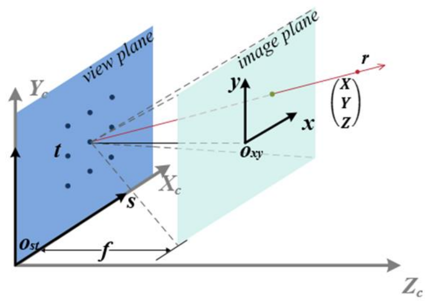


Figure 1. Two-parallel plane model for light field.

In this matrix, $(k_i, k_j, k_v, k_u, u_0, v_0)$ are intrinsic parameters of a light field camera. The light field essentially represents a set of ray,[22] in order to get more constraints for pose estimation between these rays, the Plücker coordinates is used to characterize the rays in this paper,[21,25] as shown in Eq. 2.

$$\begin{cases} \boldsymbol{m} = (s, t, 0)^\top \times (x, y, 1)^\top = (t, -s, sy - tx)^\top \\ \boldsymbol{q} = (x, y, 1)^\top \end{cases}, \tag{2}$$

where the pair of vectors $(\boldsymbol{m}^\top, \boldsymbol{q}^\top)^\top$ represent the ray mathematically. $\boldsymbol{m}$ is the moment vector and $\boldsymbol{q}$ is the direction vector. In this paper, we will model the process of sampling for light field camera in Plücker coordinates. Substituting Eq. 1 into Eq. 2, based on ray-space intrinsic matrix $\boldsymbol{K}$ the relationship between the ray $\mathcal{L} = (\boldsymbol{n}^\top, \boldsymbol{p}^\top)^\top$ captured by light field camera and the normalized undistorted physical ray $\mathcal{L}^c = (\boldsymbol{m}^\top, \boldsymbol{q}^\top)^\top$ can be established in the Plücker coordinates, $i, e.$,

$$\begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{q} \end{bmatrix} = \underbrace{\begin{bmatrix} k_j & 0 & 0 & 0 & 0 & 0 \\ 0 & k_i & 0 & 0 & 0 & 0 \\ -k_j u_0 & -k_i v_0 & k_i k_v & 0 & 0 & 0 \\ 0 & 0 & 0 & k_u & 0 & u_0 \\ 0 & 0 & 0 & 0 & k_v & v_0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{=:\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{ij} & \\ & \boldsymbol{K}_{uv} \end{bmatrix}} \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{p} \end{bmatrix}, \tag{3}$$

it should be noted that $\boldsymbol{p} = (u, v, 1)^\top$ represents the direction of ray in the sub-aperture image coordinates and $\boldsymbol{n} = (i, j, 0)^\top \times (u, v, 1)^\top$ is the moment of ray. After that, because all light field cameras recorded the scene rays is in the world coordinates, using Eq. 4, each pair of light field camera coordinates can be related by a rotation matrix $\boldsymbol{R} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3) \in SO(3)$ and a translation vector $\boldsymbol{t} = (t_x, t_y, t_z)^\top \in \mathbb{R}^3$

$$\boldsymbol{X}_1^c = \boldsymbol{R}\boldsymbol{X}_2^c + \boldsymbol{t} \tag{4}$$

Substituting Eq. 4 into Eq. 3, the correlation between rays in the light field coordinates system is

$$\begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{p}_1 \end{bmatrix} = \boldsymbol{K}^{-1} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{E} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{R} \end{bmatrix} \boldsymbol{K} \begin{bmatrix} \boldsymbol{n}_2 \\ \boldsymbol{p}_2 \end{bmatrix}, \tag{5}$$

where $\boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R}$ are essential matrix. $\mathcal{L}_1 = (\boldsymbol{n}_1^\top, \boldsymbol{p}_1^\top)^\top$ and $\mathcal{L}_2 = (\boldsymbol{n}_2^\top, \boldsymbol{p}_2^\top)^\top$ is the Plücker coordinates ray in different light field camera coordinates, respectively.

Subsequently, Eq. 6 gives a necessary and sufficient condition for the intersection of two lines in the same Plücker coordinates.

$$\boldsymbol{n}_1^\top \boldsymbol{p}_2 + \boldsymbol{p}_1^\top \boldsymbol{n}_2 = 0 \tag{6}$$

Substituting Eq. 5 into Eq. 6, the corresponding ray sets ($\{\mathcal{L}_1\} \leftrightarrow \{\mathcal{L}_2\}$) about fundamental matrix $\boldsymbol{F}$ satisfies

$$\left[\begin{array}{cc} \boldsymbol{n}_1^\top & \boldsymbol{p}_1^\top \end{array}\right] \underbrace{\left[\begin{array}{cc} \boldsymbol{K}_{ij} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{K}_{uv} \end{array}\right]^\top \left[\begin{array}{cc} \boldsymbol{0}_{3\times3} & \boldsymbol{R} \\ \boldsymbol{R} & \boldsymbol{E} \end{array}\right] \left[\begin{array}{cc} \boldsymbol{K}_{ij} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{K}_{uv} \end{array}\right]}_{\boldsymbol{F}} \left[\begin{array}{c} \boldsymbol{n}_2 \\ \boldsymbol{p}_2 \end{array}\right] = 0 \tag{7}$$

Then, the fundamental matrix $\boldsymbol{F}$ containing an unknown scale factor λ can be obtained by solving the following linear euqations

$$\left[\begin{array}{cc} \boldsymbol{n}_1^\top & \boldsymbol{p}_1^\top \end{array}\right] \otimes \left[\begin{array}{cc} \boldsymbol{n}_2^\top & \boldsymbol{p}_2^\top \end{array}\right] \vec{\boldsymbol{F}} = 0, \tag{8}$$

where $\otimes$ is a direct product operator and $\vec{\boldsymbol{F}}$ is column vector (36×1) stretched the fundamental matrix $\boldsymbol{F}$.

Finally, considering $\boldsymbol{R}$ is a rotation matrix $\boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}$, we then combine Eq. 7 and have the pose estimation for LF-SLAM $[\boldsymbol{R}|\boldsymbol{t}]$

$$\lambda \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{F}_{12} \\ \boldsymbol{F}_{21} & \boldsymbol{F}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{K}_{ij}^\top \boldsymbol{R} \boldsymbol{K}_{uv} \\ \boldsymbol{K}_{uv}^\top \boldsymbol{R} \boldsymbol{K}_{ij} & \boldsymbol{K}_{uv}^\top \boldsymbol{E} \boldsymbol{K}_{uv} \end{bmatrix} \Rightarrow \begin{array}{l} \lambda \boldsymbol{F}_{12} = \boldsymbol{K}_{ij}^\top \boldsymbol{R} \boldsymbol{K}_{uv} \\ \lambda \boldsymbol{F}_{21} = \boldsymbol{K}_{uv}^\top \boldsymbol{R} \boldsymbol{K}_{ij} \\ \lambda \boldsymbol{F}_{22} = \boldsymbol{K}_{uv}^\top [\boldsymbol{t}]_\times \boldsymbol{R} \boldsymbol{K}_{uv} \end{array} \tag{9}$$

$$\begin{aligned} &\lambda = \frac{1}{2}\left[\frac{1}{\left|\boldsymbol{K}_{ij}^{-\top}\boldsymbol{F}_{12}\boldsymbol{K}_{uv}^{-1}\right|} + \frac{1}{\left|\boldsymbol{K}_{uv}^{-\top}\boldsymbol{F}_{21}\boldsymbol{K}_{ij}^{-1}\right|}\right] \\ &\boldsymbol{R} = \lambda \boldsymbol{K}_{ij}^{-\top}\boldsymbol{F}_{12}\boldsymbol{K}_{uv}^{-1} \\ &[\boldsymbol{t}]_\times = \lambda \boldsymbol{K}_{uv}^{-\top}\boldsymbol{F}_{22}\boldsymbol{K}_{uv}^{-1}\boldsymbol{R}^\top \end{aligned} \tag{10}$$

In this formula derivation, the intrinsic matrix $\boldsymbol{K}$ is known. $|\cdot|$ denotes the determinant of matrix and $\boldsymbol{F}_{mn}$ is a 3×3 partitioned matrix.

3.2 Nonlinear Optimization for Camera Pose and 3D Points

In this section, the initial camera pose and 3D points computed by the linear method will be refined via nonlinear optimization. Different with traditional re-projection error, we define a ray-ray cost function to optimize the camera pose. In this nonlinear solution, we are seeking extrinsic parameter $[\boldsymbol{R}|\boldsymbol{t}]$ to minimize the geometrical distance between the ray set $\mathcal{L}^c = (\boldsymbol{m}^\top, \boldsymbol{q}^\top)^\top = \boldsymbol{K}(\boldsymbol{n}^\top, \boldsymbol{p}^\top)^\top$ on the calibrated light field camera coordinates and its estimate value $\hat{\mathcal{L}}^c$ as shown in Eq. 11.

$$\sum d\left(\mathcal{L}_1^c(\boldsymbol{K}, \mathcal{L}_1), \hat{\mathcal{L}}_1^c(\boldsymbol{K}, \boldsymbol{R}\boldsymbol{t}, \mathcal{L}_2)\right) + d\left(\mathcal{L}_2^c(\boldsymbol{K}, \mathcal{L}_2), \hat{\mathcal{L}}_2^c(\boldsymbol{K}, \boldsymbol{R}^\top, -\boldsymbol{R}^\top\boldsymbol{t}, \mathcal{L}_1)\right) \quad , \tag{11}$$

where $d(\mathcal{L}_1^c, \mathcal{L}_2^c) = \frac{|\boldsymbol{m}_1^\top \boldsymbol{q}_2 + \boldsymbol{q}_1^\top \boldsymbol{m}_2|}{\|\boldsymbol{q}_1 \times \boldsymbol{q}_2\|}$ is the geometric distance between two rays under Plücker coordinates.

After that, we state a point-ray cost function to optimize the 3D points. This function purpose is to minimize the geometrical distance between the 3D point set and ray set. According to Eq. 12, $\boldsymbol{X}_1^c$, $\hat{\mathcal{L}}_1^c$ are the 3D point set and the estimate value of ray set in light field camera coordinates ,respectively. $\mathcal{L}_2^c$, $\hat{\boldsymbol{X}}_2^c$ are the ray set and estimate value of 3D point set in light field camera coordinates , respectively.

$$\sum d\left(\boldsymbol{X}_1^c, \hat{\mathcal{L}}_1^c(\boldsymbol{K}, \boldsymbol{R}\boldsymbol{t}, \mathcal{L}_2)\right) + d\left(\hat{\boldsymbol{X}}_2^c(\boldsymbol{X}_1^c, \boldsymbol{R}^\top, -\boldsymbol{R}^\top\boldsymbol{t}), \mathcal{L}_2^c(\boldsymbol{K}, \mathcal{L}_2)\right) \quad , \tag{12}$$

where $d(\boldsymbol{X}^c, \mathcal{L}^c) = [\boldsymbol{q}]_\times \left\{[\boldsymbol{X}^c]_\times - \boldsymbol{m}\right\}$ denotes the geometric distance between 3D points and rays under Plücker coordinates.

In this paper, the Levenberg-Marquardt[26] algorithm is used for nonlinear optimization. The complete procedure of LF-SLAM is given in Algorithm 1.

Algorithm 1 LF-SLAM

Input:

N group of light field date $LF(i, j, u, v,)$, N group of depth value Depth(u, v), Intrinsic matrix $\boldsymbol{K}$.

Output:

Position $\boldsymbol{R}_i, \boldsymbol{t}_i$ $(1 <= i <= N)$, 3D point $\boldsymbol{X}$.

```
Initialize the first frame
Get ray-space features
Global feature F(L_i^c, X_i^c) by Eq. 3
for j = 2 to N do
    Get and match ray-space features [F_{i-1}, F_i]
    Calculate L_i^c by Eq. 3
    Calculate X_i, R_i, t_i by Eq. 7,10
    Update global feature F
    Loop
end for
for j = 2 to N do
    Optimize X_i, R_i, t_i by Eq. 11,12
end for
Mapping
```

4. EXPERIMENTAL RESULTS

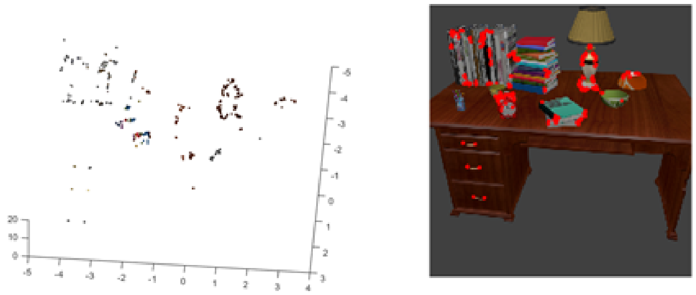
In the experiment, we simulate a light field camera to obtain different poses whose intrinsic parameters are listed in Table 1. In this paper, the system used in this experiment is Ubuntu14.04, PCL point cloud library and G2O general graph optimization library. The simulation scene data used in this experiment is built by Blender3D modeling software.There are 5×5 view points for each pose in the living room data and 7×7 view points in the desk data.

Table 1. Light field camera intrinsic parameters

k_i	k_j	k_u	k_v	u_0	v_0
0.0347826	0.0347826	0.00165631	0.00165631	−0.4579710	−0.4579482

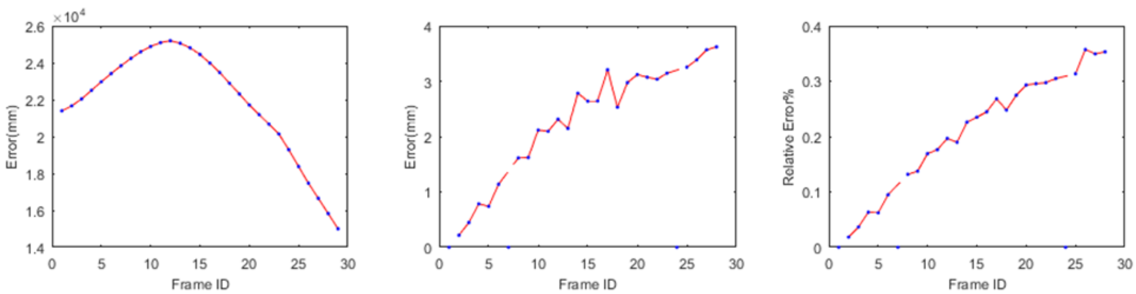
According to Algorithm 1, we did initialization and nonlinear optimization the pose on the desk data. Let the light field camera pose of the first frame be the world coordinates. This coordinates is the reference coordinates system for all the subsequent data. Fig. 2 shows an initial feature map. Fig. 3 (a) is the depth estimation error and Fig. 3 (b)(c) is ray re-projection absolute error and ray re-projection relative error on simulated data with different poses, respectively. The ray re-projection relative error is defined as dividing absolute error by the true value of related ray re-projection. According to Fig. 3, the ray re-projection error increases with the increase of the number of poses, nevertheless, the rate of error increase is decreases in the second half, what is more, there are positive correlations between ray re-projection error and depth estimation error. Fig. 4 shows the pose estimation error on the desk data, Fig. 4 (a) and Fig. 4 (b) show the absolute error and the error varition rate of pose estimation, respectively. The error varition rate is defined as $\Delta Error = E_i^{pose} - E_{i-1}^{pose}$, where E_i^{pose} indicates the absolute error for the *i-th* frame. The change law of pose estimation error is consistent with Fig. 3.

The light field camera movement locus for desk data is shown on Fig. 5, the green movement locus is obtained by proposed method and red movement locus is the ground truth. Fig. 5 shows that the estimation of camera motion is relatively accurate by proposed method, furthermore, the subsequent poses can estimate the camera poses accurately in the case of frame skip and reduce the error gradually, which further verifies the effectiveness of nonlinear optimization. Finally, it is vital to accurately reconstruct 3D points. In this section, combining with depth map and the pose estimation data for each frame, the 3D reconstruction result for living room data and



(a) (b)

Figure 2. (a) initial feature map of desk data; (b) corresponding features are displayed in the center view.



(a) (b) (c)

Figure 3. Depth and ray re-projection error map on desk data. (a) depth estimation error; (b) ray re-projection absolute error; (c) ray re-projection relative error.

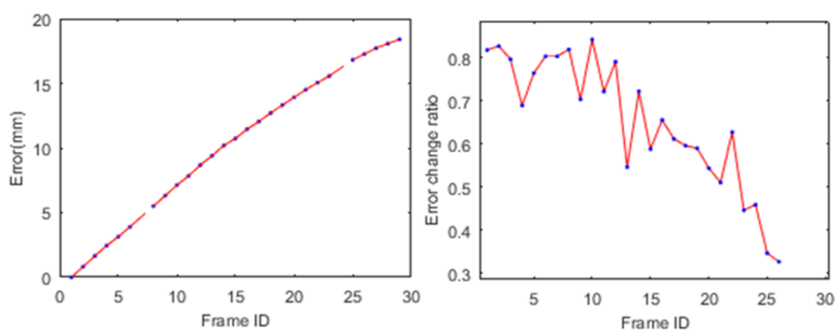


Figure 4. Pose estimation error on the desk data. (a) absolute error ; (b) the error change ratio.

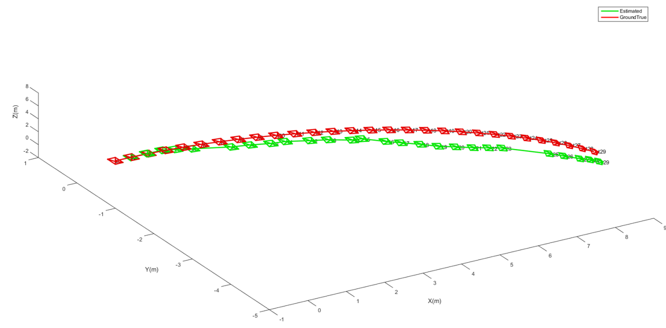


Figure 5. Movement locus for desk data.

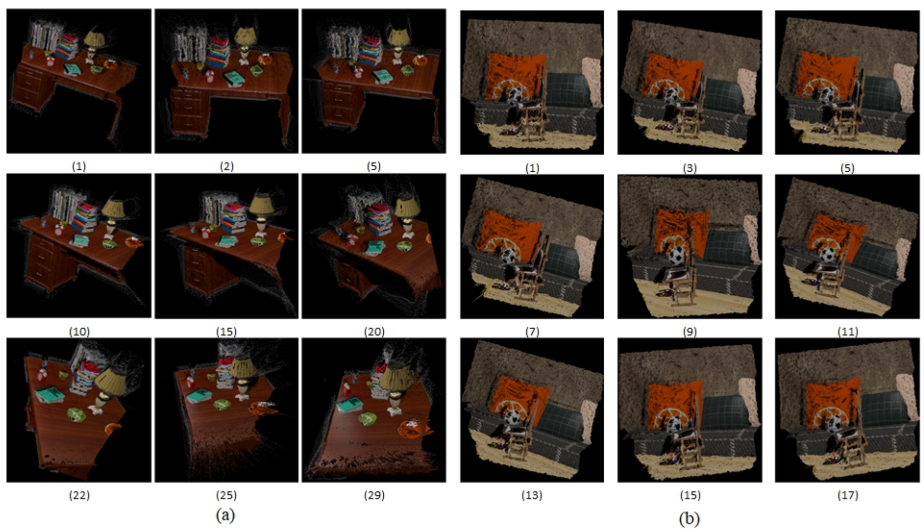


Figure 6. 3D reconstruction results from single light field frame. (a) desk data; (b) living room data.

desk data is given. Fig. 6 is a light field date 3D reconstruction result for single frame, the subscript of the figure is the corresponding frame number. As shown in Fig. 6, due to the error in the depth value there are a lot of noise, error points and some missing values in the single pose reconstruction result. Therefore, multi-pose data is used to optimize the reconstruction results in this paper. Fig. 7 is the reconstruction results for desk data (Fig. 7 (a)) and living room data (Fig. 7 (b)). It is obvious from the Fig. 7 noise is greatly reduced, holes and missing parts are well complemented in the result of multi-pose reconstruction. This results further prove the correctness of proposed method.

5. CONCLUSIONS

In this paper, according to the projection relationship between the rays in 3D space, we exploit a novel light field SLAM (LF-SLAM) based on ray-space projection model. We first initialize camera motion by ray-space projection model. After that, a novel ray-ray cost function and a point-ray cost function are established to nonlinearly optimize the camera pose and 3D points. Finally, the experiments on qualitative and quantitative comparisons verify the effectiveness and robustness of the proposed LF-SALM.

Acknowledgement

The work was supported by NSFC under Grants No.61531014 and No.61801396.

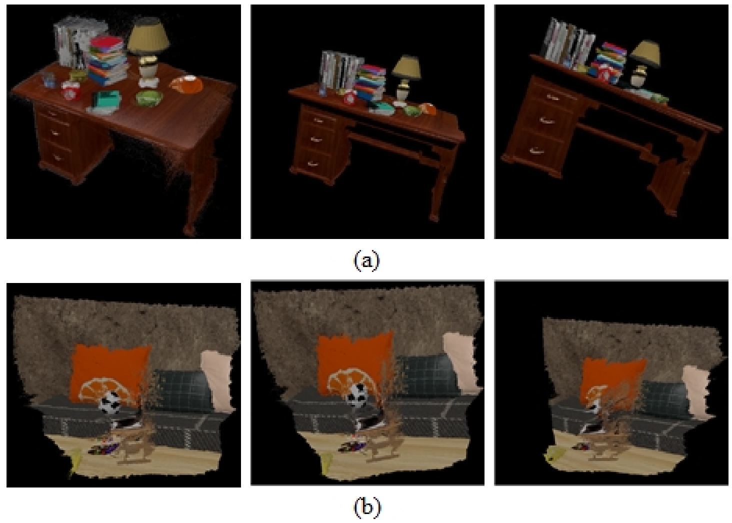


Figure 7. 3D reconstruction results from multiple light field data. (a) desk data; (b) living room data.

REFERENCES

[1] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis & Machine Intelligence* (6), 1052–1067 (2007).

[2] Klein, G. and Murray, D., "Parallel tracking and mapping for small ar workspaces," in [*Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*], 1–10, IEEE Computer Society (2007).

[3] Herrera, C. D., Kim, K., Kannala, J., Pulli, K., and Heikkilä, J., "Dt-slam: Deferred triangulation for robust slam," in [*2014 2nd International Conference on 3D Vision*], **1**, 609–616, IEEE (2014).

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

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

[6] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D., "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics* **31**(5), 1147–1163 (2015).

[7] Concha, A. and Civera, J., "Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence," in [*2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], 5686–5693, IEEE (2015).

[8] Jose Tarrio, J. and Pedre, S., "Realtime edge-based visual odometry for a monocular camera," in [*Proceedings of the IEEE International Conference on Computer Vision*], 702–710 (2015).

[9] Engel, J., Koltun, V., and Cremers, D., "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence* **40**(3), 611–625 (2017).

[10] Geiger, A., Ziegler, J., and Stiller, C., "Stereoscan: Dense 3d reconstruction in real-time," in [*2011 IEEE Intelligent Vehicles Symposium (IV)*], 963–968, Ieee (2011).

[11] Mur-Artal, R. and Tardós, J. D., "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics* **33**(5), 1255–1262 (2017).

[12] Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J., and Berlles, J. J., "S-ptam: Stereo parallel tracking and mapping," *Robotics and Autonomous Systems* **93**, 27–42 (2017).

[13] Huai, J., Toth, C. K., and Grejner-Brzezinska, D. A., "Stereo-inertial odometry using nonlinear optimization," in [*International Technical Meeting of the Satellite Division of the Institute of Navigation*], (2015).

[14] Gomez-Ojeda, R., Briales, J., and Gonzalez-Jimenez, J., "Pl-svo: Semi-direct monocular visual odometry by combining points and line segments," in [*2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], 4211–4216, IEEE (2016).

[15] Strasdat, H., Davison, A. J., Montiel, J. M., and Konolige, K., "Double window optimisation for constant time visual slam," in [*2011 International Conference on Computer Vision*], 2352–2359, IEEE (2011).

[16] Ng, R., "Lytro redefines photography with light field cameras," (2017).

[17] Levoy, M. and Hanrahan, P., "Light field rendering," in [*Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*], 31–42, ACM (1996).

[18] Liang, C.-K. and Ramamoorthi, R., "A light transport framework for lenslet light field cameras," *ACM Transactions on Graphics (TOG)* **34**(2), 16 (2015).

[19] Bok, Y., Jeon, H.-G., and Kweon, I. S., "Geometric calibration of micro-lens-based light field cameras using line features," *IEEE transactions on pattern analysis and machine intelligence* **39**(2), 287–300 (2016).

[20] Zhang, Q., Zhang, C., Ling, J., Wang, Q., and Yu, J., "A generic multi-projection-center model and calibration method for light field cameras," *IEEE transactions on pattern analysis and machine intelligence* (2018).

[21] Zhang, Q., Ling, J., Wang, Q., and Yu, J., "Ray-space projection model for light field camera," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 10121–10129 (2019).

[22] Ng, R. et al., [*Digital light field photography*], stanford university Stanford (2006).

[23] Guo, X., Yu, Z., Kang, S. B., Lin, H., and Yu, J., "Enhancing light fields through ray-space stitching," *IEEE transactions on visualization and computer graphics* **22**(7), 1852–1861 (2015).

[24] Zhang, C., Ji, Z., and Wang, Q., "Rectifying projective distortion in 4d light field," in [*2016 IEEE International Conference on Image Processing (ICIP)*], 1464–1468, IEEE (2016).

[25] Jia, Y.-B., "Plücker coordinates for lines in the space," (2018).

[26] Madsen, K., Nielsen, H. B., and Tingleff, O., "Methods for non-linear least squares problems," (1999).