Vis Comput
DOI 10.1007/s00371-017-1435-0

ORIGINAL ARTICLE

Handling pure camera rotation in semi-dense monocular SLAM

Yao Zhou[1] · Feihu Yan[1] · Zhong Zhou[1]

Abstract In this paper, we present a method for semi-dense monocular simultaneous localization and mapping (SLAM) that is capable of dealing with pure camera rotation motion which brings forward a severe challenge for current direct (featureless) monocular SLAM approaches. A probabilistic depth map model built on Bayesian estimation is combined with the main framework of the state-of-the-art direct method LSD-SLAM. Using this model, both rotation-only and general camera motions could be tracked, and a consistent depth map could be built in real-time. Experimental results demonstrate the outstanding performance of the proposed system.

Keywords Semi-dense visual SLAM · Rotation-only camera motion · Direct method

1 Introduction

Benefited from the rapid development of virtual reality (VR) and augmented reality (AR) devices and applications, real-time simultaneous localization and mapping (SLAM) has been gaining increasing popularity as an essential part of AR and VR researches [1–3] in the last two decades. SLAM techniques can be divided into different classes according to different sensors like lasers, sonar or cameras. With a monocular camera, the cheapest and smallest sensor module, visual monocular SLAM algorithms [4–14] have made significant progress, and feature-based techniques have been consolidated and prevalent in the past decades. Recently, direct approaches are drawing more and more attentions. Contrast to feature-based approaches which extract and triangulate features on the images, direct approaches track camera motions and reconstruct the environment directly over pixel intensities on the whole image. This provides substantially more information about the environment, which can be invaluable for robotics or augmented reality applications.

Though with more potential applications, direct approaches still have many restrictions on the camera motion. Particularly in rotation-only camera motion, existing direct semi-dense SLAM systems could hardly estimate and update the depth map and finally cause tracking failed.

In this paper, we build on the main framework of LSD-SLAM [8] and the probabilistic depth map model [15], to design a semi-dense monocular SLAM system suitable for rotation motion. More specifically, we model the depth of every pixel as a distribution that mixes a good measurement (normally distributed around the true depth) and an unknown measurement (uniformly distributed in an interval which is supposed to contain the depth range). As new frames arrive, we regard these frames as new observations for the depth of their reference keyframe and compute the Bayesian estimation for the real depth of the keyframe and estimate the probability of satisfactory ones. Choosing this model, depth map can still be created in rotation-only camera motion. Both general and rotation-only camera motion can be tracked, and a semi-dense map could be reconstructed at last, as shown in Fig. 1.

The remainder of this paper is organized as follows: Section 2 presents related works. Section 3 introduces the probabilistic depth map, including the Bayesian model of the map as well as the map update, propagation and regularization steps. Section 4 introduces how to track new frames

✉ Feihu Yan
yfhmail@163.com

Zhong Zhou
zz@buaa.edu.cn

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

Published online: 01 September 2017

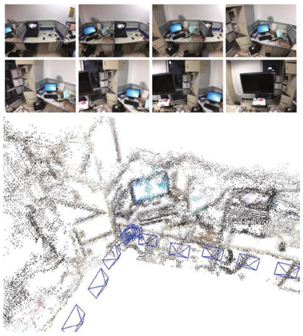

Fig. 1 The reconstructed semi-dense map and estimated keyframe poses for sequence 1 of our system. Both general and rotation-only motions can be tracked successfully

with the depth map. Experimental results are illustrated in Sect. 5, and Sect. 6 draws a conclusion.

A previous version of our work was presented in [16]. In this paper, we add descriptions in detail and perform more experiments.

2 Related work

A large variety of SLAM systems have been proposed in the past decades, which can be intuitively divided into two classes: feature-based and direct approaches. A feature-based approach estimates camera poses and reconstructs maps by extracting and tracking a sparse set of image features from successive frames, while a direct approach minimizes the photometric error directly over pixel intensities and performs dense or semi-dense reconstruction. Works belonging to the former and the latter class include [4–6] and [7,8,13,17,18], respectively.

The initial approaches of feature-based monocular SLAM systems are mostly based on filter methods. Davison et al. [4] firstly presented a real-time monocular SLAM system called MonoSLAM, employing EKF-based probabilistic estimation to calculate camera poses and build a sparse map of features. Modern feature-based approaches [5,6] are based on keyframes [19]. Optimization methods such as bundle adjustment (BA) [20] could be operated in these systems. Klein and Murray [5] suggested a widely popular framework, parallel tracking and mapping (PTAM), which splits camera tracking and mapping into two parallel threads and performs optimization over selected frames applying BA methods. Murartal et al. [6] designed a novel monocular SLAM system, ORB-SLAM. Built on the main ideas of PTAM, the system has fixed many limits such as loop closing and relocalization and becomes one of the most representative feature-based SLAM techniques.

Recently, as the performance of computer hardware has been incredibly improved, multiple kinds of direct approaches have been put forward. Newcombe et al. [13] presented a dense SLAM system which generates smooth depth estimate by a non-convex optimization process. This system needs GPU to enhance the processing power. The first large-scale direct monocular SLAM method is LSD-SLAM, a real-time direct monocular SLAM framework, proposed by Engel et al. [7,8]. The system employs a direct tracking method towards keyframes and a probabilistic filtering solution to build large-scale semi-dense maps. It is impressive that this system has real-time capability on CPUs without GPU acceleration. Then, Caruso et al. [17,18] extended this framework to an omnidirectional camera model and a stereo camera model, respectively.

There are also many systems using a combination of feature-based methods and direct approaches, such as SVO [14], which is proposed by Forster et al. They use direct methods to estimate feature correspondences and feature-based methods to refine camera poses.

All the mentioned SLAM works seek for robust real-time performance; however, tracking is usually failed in multiple situations such as pure rotation. Handling rotation-only camera motion has always been one severe challenge for SLAM. Several algorithms have been proposed to explicitly address this problem.

Gauglitz et al. [9] presented a keyframe-based real-time approach which differentiates general and rotation-only camera motions between keyframe pairs. In the latter case, Pirchheim et al. [10] proposed a scheme with the basic idea of combining 6DOF and panoramic SLAM, the regional panorama maps registered in a global 3D map to handle pure rotation camera movements. Herrera et al. [11] presented a real-time visual SLAM system that tracks the features locally and incrementally and delays triangulation of the matched 2D features between keyframes until sufficient baseline has been satisfied.

Theoretically treating translation motion and rotation motion differently should be fine. However, the two kinds of camera motion could be inextricably linked in practice. On the other hand, few approaches of direct method SLAM have been proposed to handle this degenerate rotation-only camera motion. And likewise, this is the major motive of this paper.

In this work, we propose a direct monocular SLAM that combines a probabilistic depth map model based on Bayesian estimation with the main framework of LSD-SLAM to deal with not only general camera motions but also rotation-only motions.

3 Probabilistic depth map based on Bayesian estimation

In LSD-SLAM [8], the system uses an extended Kalman filter to refine the depth map. More specifically, when

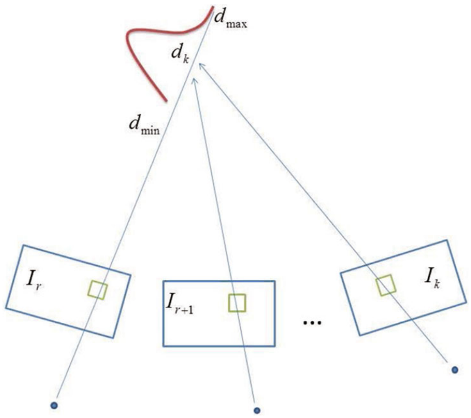

Fig. 2 Bayesian estimation of the depth. A sequence of estimated depths is regarded as independent observations of the true depth

a frame is chosen to be a keyframe, the system estimate the depth of all pixels which have a non-negligible image gradient in new images, and each estimate is represented as a Gaussian probability distribution and used to refine the depth map of the keyframe. The potential meaning is that each estimate would be treated as a good measurement.

In practice, however, there are always numerous inevitable bad measurements. If we could separate bad measurements from good measurements, the depth estimate would be more accurate with restricted iterations. In particular for real-time keyframe-based SLAM system, the reference keyframes could be generated frequently to maintain effective tracking when severe camera motion occurs, which also means observations for one depth would be restricted while introducing more noisy estimations.

Considering this situation, one depth model which is less affected by outliers could be more suitable for our system. Therefore, we model the estimated depth d_k of each pixel according to [15] with a distribution that mixes a good measurement (normally distributed around the true depth $\hat{d}$) and an unknown measurement (uniformly distributed in an interval $[d_{\min}, d_{\max}]$):

$$p(d_k|\pi) = \pi \mathcal{N}\left(d_k|\hat{d}, \tau_k^2\right) + (1-\pi)\mathcal{U}(d_k|d_{\min}, d_{\max}) \quad (1)$$

where π and τ_k^2 are the probability and the variance of a good measurement in k-th frame. Note that we use d to denote the inverse depth, which is different with [15]. And this model is illustrated in Fig. 2.

As derived in [15], the posterior of the Bayesian estimation for d can be approximated by the product of a Gaussian distribution for the depth and a Beta distribution for the probability of good measurement:

$$q\left(\pi, \hat{d}|a_k, b_k, \mu_k, \sigma_k^2\right) = \text{Beta}\left(\pi \mid a_k, b_k\right)\mathcal{N}\left(\hat{d}|\mu_k, \sigma_k^2\right) \quad (2)$$

where a_k and b_k are the parameters of Beta distribution.

More details of this model are presented in [15] and similarly it is also used in SVO [14].

Similar to LSD-SLAM, the main mapping process in our system contains four parts: depth map initialization, depth map update, depth map propagation and depth map regularization, while all steps have been modified to combine with the applied depth map model. Depth map initialization step uses a random method to initialize the depth map and gives it a large variance. Depth map update step computes the depth observations in each frame and updates the depth map of the current keyframe. Depth map propagation step creates a new keyframe when the current frame is too far away from the existing depth map and propagates depth map from the old keyframe into the new one. Depth map regularization step is executed after the update step. It computes the smoothed depth for stereo searching and tracking. Overview of the mapping process is visualized in Fig. 3.

3.1 Depth map initialization

Instead of estimating the relative pose between two or more frames to triangulate initial map in traditional monocular visual SLAM systems, LSD-SLAM [8] uses an initialization method that initialize the first keyframe with a random depth map with large variance. In practice, these initialized depths are always outliers. Since we choose the Bayesian model [15] in our system, which could naturally separate good measurements and unknown measurements, we could make full use of this model in the depth map initialization step.

In detail, as visualized in Fig. 4, we initiate each pixel in the depth map with a high expectation of unknown measurement and this step will generate a random depth and large variance. After several subsequent frames, the depth map could be upgraded to a correct depth configuration using the Bayesian estimation mentioned above.

3.2 Depth map update

When the camera pose of a new frame has been estimated, depth map update step would be used to update the depth map of the reference keyframe. For every pixel with non-negligible gradient in the keyframe, a search method which matches the pixels intensity along the epipolar line on the current frame is performed. In order to improve the search efficiency, the search interval $d \pm l\ (E_\pi, \sigma)$ is limited by the prior info of the pixel, and $l\ (E_\pi, \sigma)$ is defined as

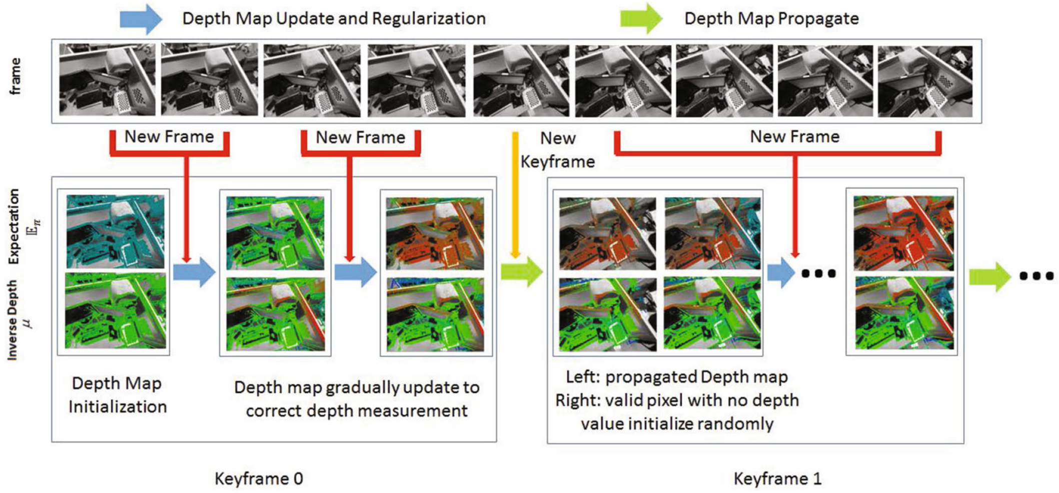
Depth Map Update and Regularization
Depth Map Propagate
frame
New Frame
New Frame
New Keyframe
New Frame
Expectation E_π
Inverse Depth μ
Depth Map Initialization
Depth map gradually update to correct depth measurement
Left: propagated Depth map
Right: valid pixel with no depth value initialize randomly
Keyframe 0
Keyframe 1

Fig. 3 Mapping process of our approach. In the first keyframe, a depth map is randomly initialized. As new frames captured, depth observations are calculated by stereo matching, while the depth map is updated at the same time. When the depth map update step finished, the regularization step will be carried out to get a smoothed depth for tracking and stereo searching. When camera moves far away from the current keyframe, a new keyframe will be created and the depth map of the previous keyframe should be propagated into it. In new depth map, valid pixels with no assigned depth measurement would be initialized with a random method. Then, the new depth map will be updated and regularized iteratively again

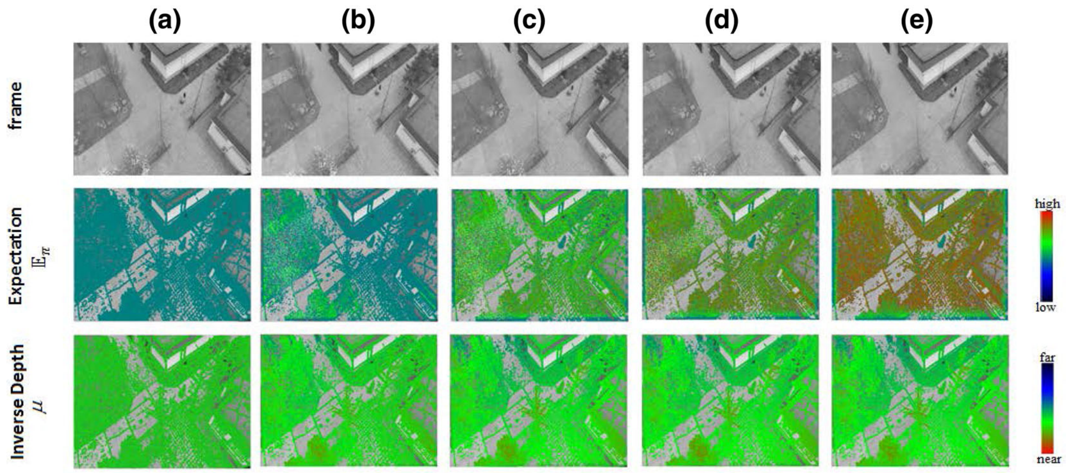

Fig. 4 With several input image frames, a random initialization depth map can be upgraded to a correct depth configuration. In the front several frames (col. **a**–**c**), the inverse depth map is converged to a correct value and expectation map keep rising. Then, with more input frames (col. **c**–**e**), inverse depth map remained steady and expectation map rised to a high level

$$l(E_\pi, \sigma) = 2\pi_i \sigma_d + (1 - E_\pi)\sigma_{\max} \tag{3}$$

where the parameter E_π is the expected value for the probability of a good measurement, and it is controlled by the Beta distribution in Eq. (2). In other words, this parameter is controlled by pixel parameter a and b: $E_\pi = \frac{a}{a+b}$. Parameter $\sigma_{\max}$ is a constant which represents 99% of the probability inverse depth lies in the range $[d_{\min}, d_{\max}]$ by the Gaussian distribution. Parameter σ_d is the inverse depth variance of one pixel which is estimated by previous observations (Figs. 5, 6).

Then, we need to estimate the uncertainty of the inverse depth. The method applied in [7,8] which considers both photometric and geometric disparity errors and together with the pixel to inverse depth ratio, is performed to determine the accuracy of this stereo observation. Though the three factors are designed for small camera rotation hypothesis in [7,8], we note that if the reference keyframe could be generated more frequently as rotation-only camera motion occurs, the estimate method could also be reliable. We refer to the original work in [7] for more details of this estimate method.

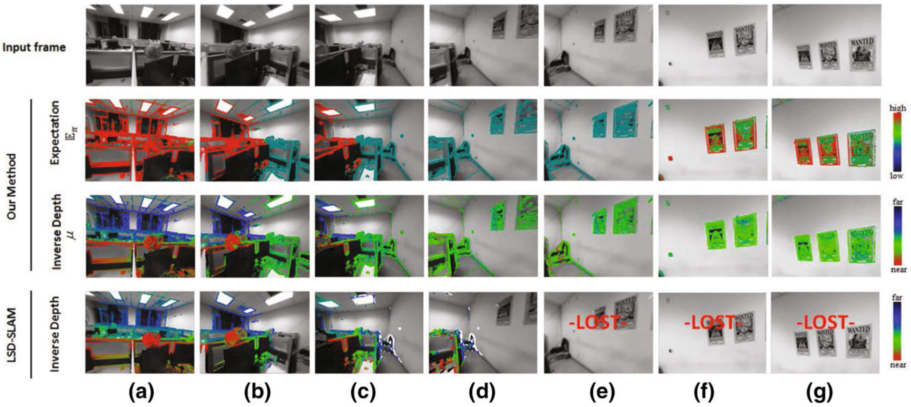

Fig. 5 Keyframe sequences during the period of rotation-only motion in our system and LSD-SLAM. These sequences are selected in image sequence 1 from image 650 to image 900. Both systems work well in general motion (col. **a**). Then, pure rotation motion begins. The number of depth points in LSD-SLAM keeps decreasing (col. **b**–**d**), and the system fails in tracking at last. While our system could deal with new pixels. These new pixels remain low E_π during rotation motion (col. **b**–**d**) because of lacking parallax but are enough for tracking. When pure rotation camera motion ends, the depth map of our system could quickly be updated to a correct depth configuration (col. **f**–**g**)

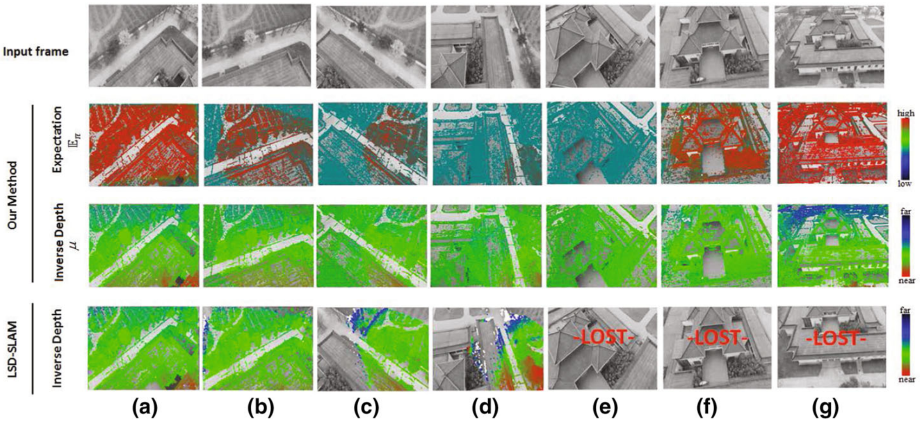

Fig. 6 Keyframe sequences in the period of rotation-only motion of our system and LSD-SLAM. The sequence is selected in image sequence 2 from image 1080 to image 1600

When the inverse depth and its uncertainty of the pixel in the current observation have been estimated, these factors will be added into the mentioned Bayesian estimation. Parameters a, b, μ and σ which correspond to the depth measurement of the pixel is then updated and would be used for tracking and mapping afterwards.

3.3 Depth map propagation

Incoming frames are evaluated to determine whether they should be added as a keyframe. Since the search method matching non-negligible gradient of pixels along the epipolar line on the current frame with the reference keyframe has been performed, it is able to determine which parts of the new frame have been tracked in the depth map. To create a new keyframe, the following main conditions should be met:

1. A given number of pixels of the new frame have not been tracked.
2. A given distance or a given angle between the current frame and the reference keyframe is reached.

Note that in the second condition, keyframes could be generated frequently in our system, which is different with LSD-SLAM.

If the camera moves far away from the current keyframe, or the rotation increases sharply, a new keyframe would be created from the recent tracked frame. Based on the estimated camera motion between the two frames, the depth map of last keyframe will be projected into the new keyframe. New inverse depth is calculated by

$$\mu_k(x_k) = Tk^{-1}(x_0, \mu_0(x_0))|_Z \tag{4}$$

where x_k is the corresponding pixel position in the new keyframe:

$$x_k = kTk^{-1}(x_0, \mu_0(x_0)) \tag{5}$$

Since keyframes could be generated frequently when pure rotation occurs, the camera rotation between the two keyframes could be assumed to be small in both general and rotation-only camera motions. Then, the new variance can be approximated by

$$\sigma_{\mu_k}^2 = \left(\frac{\mu_k}{\mu_0}\right)^4 \sigma_{\mu_0}^2 + \sigma_c^2 \tag{6}$$

where σ_c^2 is a constant which approximately corresponds to the camera motion uncertainty. And parameters a and b which correspond to the probability of good measurement are simply equal to values in the previous keyframe.

Then, the depth measurement is allocated to the closest integer pixel position. For every pixel in the new keyframe, at most one depth measurement would be allowed. If there are two depth measurements are generated for one pixel, we need to handle the collision. Let η_{th} be the thresholds on the expectation of good measurement. There are three cases:

(a) If both pixels meet the limit $E_\pi > \eta_{\text{th}}$. Then if $|\mu_1 - \mu_2| \le \sigma_1 + \sigma_2$, they will be consider to be two independent estimations of one pixel, and we would fuse them. Otherwise, the point that is closer from the camera will be retained and the farther one would be considered to be occluded, and will be removed.
(b) If only one pixel meets the limit $E_\pi > \eta_{\text{th}}$, we choose this one to be remained.
(c) If neither of the two pixels meets the limit, we will randomly choose one.

If a pixel with non-negligible gradient has no assigned depth measurement, the pixel will be initialized with a high expectation of indefinite measurement and will get a random depth and large variance. Then as new observations have been added into the Bayesian estimation, the depth measurement could be efficiently converged to the true value.

3.4 Depth map regularization

After the keyframe has been updated by subsequent new frames, one iteration regularization method will be performed to smooth the inverse depth value. In detail, we average the surrounding inverse depths with the weights of their possibility of good measurement and inverse variance. In order to preserve sharp edges, only pixels with adjacent depth will be calculated. The regularization function is defined as:

$$\mu_{\text{smooth}}(x) = \frac{\sum_{x' \in \Omega_x} \alpha g(E_\pi(x'), \sigma(x'))\mu_{\text{raw}}(x')}{\sum_{x' \in \Omega_x} \alpha g(E_\pi(x'), \sigma(x'))} \tag{7}$$

where Ω_x is the set of valid pixels around pixel x in $3 * 3$ resolution, and $g(\pi, \sigma)$ is the weighting function which will be introduced in Sect. 4. Parameter α is used to preserve sharp edge and is defined as

$$\alpha(\mu, \sigma, \mu', \sigma') = \begin{cases} 0 & \|\mu - \mu'\|_1 > 2\|\sigma - \sigma'\|_1 \\ 1 & \text{else} \end{cases} \tag{8}$$

The smoothed depth will be utilized to restrict the stereo search range (Sect. 3.2) and track new frames (Sect. 4).

4 Dense tracking based on the probabilistic depth map

The camera pose of new frame is estimated using the dense image alignment based on the depth map of the reference keyframe. As has been successfully applied in [7,21], the photometric error for a pixel is defined as

$$r_I = I_2(kTk^{-1}(x, \mu(x))) - I_1(x) \tag{9}$$

where k is the camera projection matrix and k^{-1} is the inverse. $T \in \mathrm{SE}(3)$ is a transformation matrix which represents the camera motion from the reference frame to the current frame. Since T has twelve parameters while the camera motion only has six degrees of freedom, we use Lie algebra $\xi \in \mathrm{se}(3)$ which is associated with the group SE(3). Then, the transformation matrix T can be calculated based on ξ using the exponential $T = \exp(\xi)$. $I_1(x)$ is the intensity of pixel in the reference frame, and $I_2(x)$ the intensity in the current frame.

In order to enhance robustness, we add an additional weighting term which is calculated by the probability and variance of good measurement for each valid point. The camera motion ξ^* is calculated by minimizing the energy function:

$$\xi^* = \arg\min_{\xi} \sum_{x \in \Omega} g(E_\pi(x), \sigma_i(x)) \|r_i(\xi, x)\|_\varepsilon \tag{10}$$

where $\|r_i(\xi, x)\|_\varepsilon$ is the Huber norm to penalize the outliers and increase the robustness:

$$\|r_i(\xi, x)\|_\varepsilon = \begin{cases} \frac{\|r_i(\xi,x)\|_2^2}{2\varepsilon} & \text{if } \|r_i(\xi, x)\|^2 \le \varepsilon \\ \|r_i(\xi, x)\|_1 - \frac{\varepsilon}{2} & \text{otherwise} \end{cases} \tag{11}$$

$g(\pi, \sigma)$ is the weighting function, represented as

$$g(\pi, \sigma) = \pi \frac{\sigma_{\max}^2}{\sigma^2} + (1 - \pi)\frac{\sigma_{\max}^2}{\lambda} \tag{12}$$

where λ is a constant controlling the weight item of unknown measurement. Obviously, the weight of unknown measurement should be much smaller than the good measurement and λ should be at least lager than $\sigma_{\max}^2$. In our experiments, λ is equal to $4\sigma_{\max}^2$.

When a point have a high expected value E_π for the probability of reliable measurement, the weight is mainly controlled by the measurement variance σ_i^2. With a smaller variance σ_i^2, this point can have larger weight to the minimization of energy function. Oppositely, weight will be less with a small expectation E_π or higher variance σ_i^2.

The solution to the minimization problem is computed iteratively based on the reweighted Gauss–Newton algorithm. A coarse-to-fine approach is implemented to handle larger inter-frame motions. Each new frame is first tracked on a low resolution image and depth map. The tracked pose is then used as initialization for the higher resolution. Depth map are down sampled by factors of two, using a weighted average of the inverse depth and inverse variance:

$$\mu_{l+1}(x) = \frac{\sum_{x' \in \Omega_x} g(x')\mu_l(x')}{\sum_{x' \in \Omega_x} g(x')} \tag{13}$$

$$\sigma_{l+1}(x) = \frac{\sum_{x' \in \Omega_x} g(x')\sigma_l(x')}{\sum_{x' \in \Omega_x} g(x')} \tag{14}$$

where l is the pyramid level and Ω_x is the set of valid pixel contained in pixel at the higher resolution.

5 Experimental results and discussion

The implementation of our system was extended from the main framework of LSD-SLAM. We recorded two image sequences which contain both general motion and rotation-only motion to demonstrate the additional capabilities of our system.

Image sequence 1, see Fig. 7a, captures a room-sized indoor scene and is recorded by Ipad Air2 with a fish-eye lens. And sequence 2, see Fig. 7b, is recorded by a Micro Aerial Vehicle (MAV), DJI Phantom 3. It captures the outdoor scene of a museum from the air. We processed the image sequences with both our method and LSD-SLAM. The experiment was performed on a computer equipped with a quad-core 3.5GHz CPU and 8GB of RAM. Figure 7 also shows point clouds and camera trajectories produced by our method, while LSD-SLAM fails to create complete maps.

We collect tracking and mapping statistical results in the two image sequences. While LSD-SLAM can only create submaps in different period of regular camera motion, our approach can merge these submaps separated by rotation-only camera motion into a single map and provide more restriction to loop-closing optimization. Thus we could reconstruct a lager and denser semi-dense map. In Fig. 8,

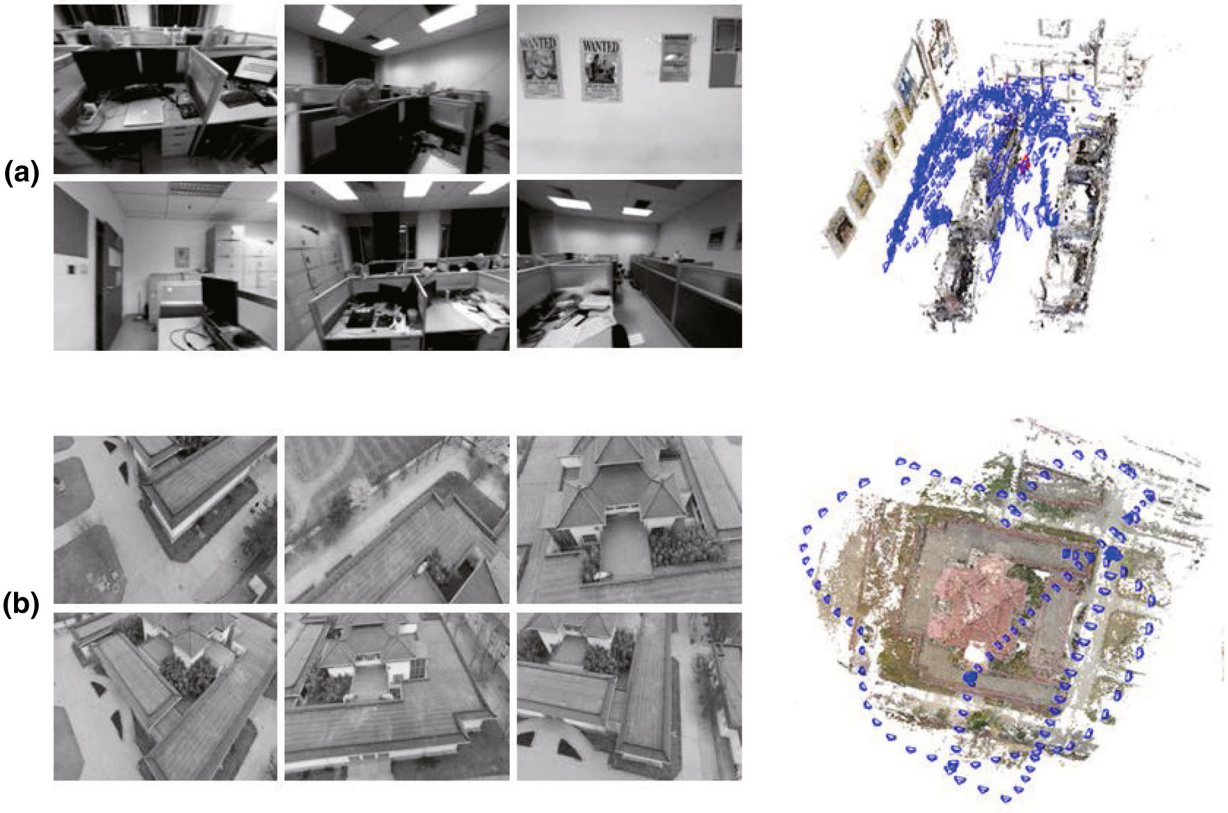

Fig. 7 Grayscale frames and reconstruction results of two image sequences. **a** Image sequence 1 captures indoor scene of a laboratory, **b** image sequence 2 captures outdoor scene of a museum

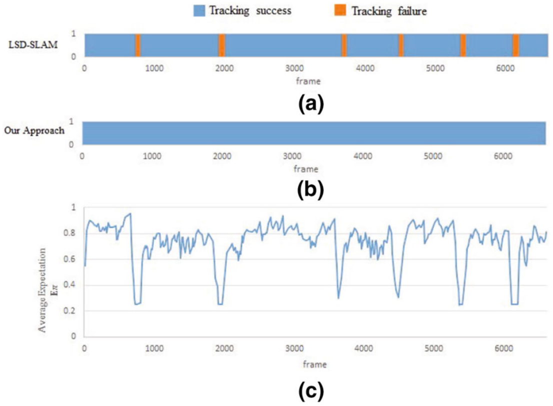

Fig. 8 Statistic results for image sequence 1 of our method and LSD-SLAM. **a** Tracking status of LSD-SLAM, **b** tracking status of our method, **c** average expectation E_π of the keyframe which can be used to evaluate the quality of camera motion

tracking and mapping statistical results for image sequence 1 is presented. In Fig. 8a, b, we observe that our method has tracked all of the frames, while LSD-SLAM fails six times (manually reset the system after a failure). All tracking failures of LSD-SLAM are caused by pure rotation camera motions, while these situations could be handled by our method. In Fig. 8c, we demonstrate the change of average expectation E_π due to different situations of camera motion. When general motion occurs, the average expectation E_π remains at a high level and is affected by the quality of camera motion. Fast camera motion and frequent keyframe change will lead to a relatively low value. While when rotation-only motion occurs, the average expectation E_π reduces to a low level, because not enough parallax is observed. But rotation-only motion can still be tracked based on the map. When the camera motion returns to be general, the average E_π returns to the formerly high level, too.

In Fig. 9, statistical results for image sequence 2 is presented. LSD-SLAM fails twice in tracking, while our method has tracked all of the frames. We also note that the broken line of average expectation E_π in Fig. 9c is smoother than line in Fig. 8c, because the camera motion in image sequence 2 is much smoother than image sequence 1.

In Figs. 5 and 6, keyframe sequences of both algorithms in the period of rotation-only camera motion are presented to intuitively show how our approach can handle rotation-only motion. Figure 5 corresponds to frame 650–900 in image sequence 1, and Fig. 6 corresponds to frame 1080–1600 in image sequence 2. Both algorithms work well in the general camera motion (col. a in Figs. 5, 6).

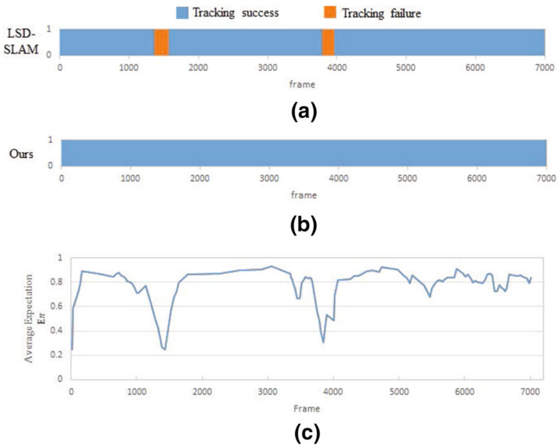

Fig. 9 Statistic results for image sequence 2 of our method and LSD-SLAM. **a** Tracking status of LSD-SLAM, **b** tracking status of our method, **c** average expectation E_π of the keyframe

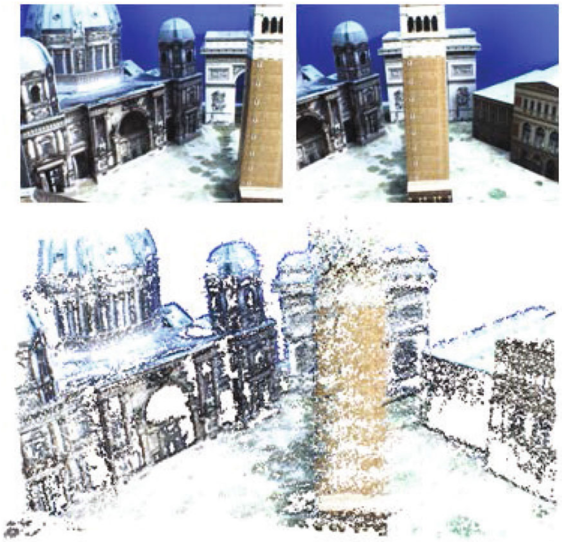

Fig. 10 Reconstruction result for the City of Slights datasets [22]. The *top* shows two frames in the dataset CS_RA_L0_BirdsView (*left to right* frame 1589 and frame 1816). The *bottom* shows the semi-dense point cloud reconstruction

When pure rotation occurs, due to the lack of parallax for mapping, LSD-SLAM cannot create new depth point and just propagate old depth point to the new keyframe. Thus, the number of valid depth points keeps decreasing (col. b–d in Figs. 5, 6) and finally the system fails in tracking. Our method initiates new non-negligible gradient pixel with low expectation E_π and random depth. The new pixel will remain low expectation E_π during rotation motion (col. b–d in Figs. 5, 6) because of lacking parallax but is enough to track the rotation-only motion. When pure rotation ends, the depth map of our method could be quickly updated to a correct depth configuration (col. e–g in Figs. 5, 6).

In order to demonstrate that our system could not only deal with the pure rotation motions but also run well in normal conditions, we evaluate the proposed approach on two widely used datasets, the City of Sights stage set [22] and the TUM RGB-D dataset [23]. Figures 10 and 11 depict the chosen frames from different views and the reconstruction results

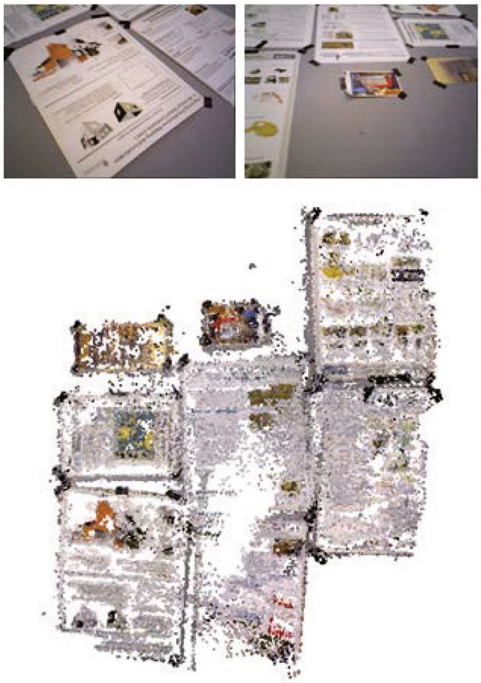

Fig. 11 Reconstruction result of sequence fr3/near [23]. The *top* shows two frames in the dataset (*left to right* frame 235 and frame 830). The *bottom* shows the semi-dense point cloud reconstruction

Table 1 Comparison of RMSE (cm) on TUM RGB-D dataset [23]

	LSD-SLAM [8]	Ours
fr1/floor	21.7	19.5
fr2/xyz	1.33	1.32
fr2/desk	3.02	3.32
fr3/office	3.96	3.43

of these two datasets which are composed of coloured semi-dense 3D points. Table 1 shows the RMSE results in four sequences of TUM RGB-D dataset [23] compared to LSD-SLAM [8], and the results are very close since there are not too many rotation-only motions in these sequences.

6 Conclusion

In this paper, we propose a real-time direct (featureless) monocular SLAM system which combines a probabilistic depth map model based on Bayesian estimation with the main framework of LSD-SLAM. The system has the capability to address rotation-only camera motion, which is always a severe challenge for current direct SLAM approaches.

The probabilistic depth map which models the depth of every pixel as a mixture of good measurement and unknown measurement is carried out, and both general and rotation-only camera motion can be handled by the computed Bayesian estimation.

Experimental results demonstrate the outstanding performance of the proposed system.

Like normal direct methods, however, our system will meet the great challenges in the presence of geometric noise or fast motion with the nature limitation of direct methods. In our future work, we would like to combine feature-based algorithms or IMU measurements to alleviate these problems.

Acknowledgements This work is supported by the National 863 Program of China under Grant No. 2015AA016403 and the Natural Science Foundation of China under Grant Nos. 61472020, 61572061, 61602223.

References

1. Reif, R., Walch, D.: Augmented & virtual reality applications in the field of logistics. Vis. Comput. **24**(11), 987–994 (2008)
2. Ott, R., Thalmann, D., Vexo, F.: Haptic feedback in mixed-reality environment. Vis. Comput. **23**(9), 843–849 (2007)
3. Wang, S.W., Cai, K., Lu, J., Liu, X., Wu, E.: Real-time coherent stylization for augmented reality. Vis. Comput. **26**, 445–455 (2010)
4. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
5. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan (2007)
6. Murartal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
7. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: IEEE International Conference on Computer Vision, Sydney, Australia (2013)
8. Engel, J., Schops, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: European Conference on Computer Vision, Zurich, Switzerland (2014)
9. Gauglitz, S., Sweeney, C., Ventura, J., Turk, M., Hollerer, T.: Live tracking and mapping from both general and rotation-only camera motion. In: IEEE International Symposium on Mixed and Augmented Reality, Atlanta, USA (2012)
10. Pirchheim, C., Schmalstieg, D., Reitmayr, G.: Handling pure camera rotation in keyframe-based SLAM. In: International Symposium on Mixed and Augmented Reality, Adelaide, SA, Australia (2013)
11. Herrera, D., Kim, C.K., Kannala, J., Pulli, K., Heikkila, J.: Dt-slam: Deferred triangulation for robust slam. In: International Conference on 3D Vision, Tokyo, Japan (2014)
12. Pizzoli, M., Forster, C., Scaramuzza, D.: RE-MODE: probabilistic, monocular dense reconstruction in real time. In: IEEE International Conference on Robotics and Automation, Hong Kong, China (2014)
13. Newcombe, R., Lovegrove, S., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: International Conference on Computer Vision (2011)
14. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation, Hong Kong, China (2014)
15. Vogiatzis, G., Hernandez, C.: Video-based, real-time multi-view stereo. Image Vis. Comput. **29**(7), 434–441 (2011)

16. Zhou, Y., Yan, F., Zhou, Z.: Probabilistic depth map model for rotation-only camera motion in semi-dense monocular SLAM. In: The 6th International Conference on Virtual Reality and Visualization, Hangzhou, China (2016)
17. Caruso, D., Engel, J., Cremers, D.: Large-scale direct SLAM for omnidirectional cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany (2015)
18. Engel, J., Stuckler, J., Cremers, D.: Large-scale direct SLAM with stereo cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany (2015)
19. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual SLAM: why filter? Image Vis. Comput. **30**(2), 65–77 (2012)
20. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) Vision Algorithms: Theory and Practice. IWVA 1999. Lecture Notes in Computer Science, vol. 1883. Springer, Berlin (2000)
21. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for RGB-D cameras. In: IEEE International Conference on Robotics and Automation, Karlsruhe, Germany (2013)
22. Gruber, L., Gauglitz, S., Ventura, J., Zollmann, S., Huber, M., Schlegel, M., Klinker, G., Schmalstieg, D., Hllerer, T.: The city of sights: design, construction, and measurement of an augmented reality stage set. In: IEEE International Symposium on Mixed and Augmented Reality, Seoul, Korea (2010)
23. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal (2012)

Yao Zhou received his M.S. degree in computer science and technology from Beihang University in 2017. His research interests include 3D visualization and SLAM.

Feihu Yan is a Ph.D. candidate in computer science at State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. He received his M.S. degree in computer science and technology from Shandong University in 2012. His research interests include 3D reconstruction and SLAM.

Zhong Zhou is a professor, Ph.D. adviser, at State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. He received his B.S. degree from Nanjing University and Ph.D. degree from Beihang University in 1999 and 2005, respectively. His main research interests include augmented virtual environment, natural phenomena simulation, distributed virtual environment, and Internet-based VR technologies. He is member of IEEE, ACM, and CCF.