ORB-SLAM MAP INITIALIZATION IMPROVEMENT USING DEPTH

Satoshi Fujimoto[†] *Zhencheng Hu*[†] *Roland Chapuis*[‡] *Romuald Aufrère*[‡]

[†] Kumamoto University, Kumamoto, Japan
[‡] Institut Pascal, Aubière, France

ABSTRACT

Map initialization and scale ambiguity are well-known challenging problems for Visual SLAM. ORB-SLAM authors [1] [2] [3] [4] have published automatic initialization and excellent Bundle Adjustment. However, scale ambiguity remains an open issue. Since depth sensors provide scale information, RGB-D SLAM improves this scale ambiguity problem. However, RGB-D SLAM depth estimation is provided only within limited areas and the measurement error usually increases with the distance. To avoid this problem, in this paper a triangulation is used on RGB feature points for getting 3D points from out of the limited area in depth. We combined both advantages of triangulation and depth. Our RGB-D ORB-SLAM system is evaluated in TUM RGB-D dataset [5] and compared with ORB-SLAM. Our analysis concludes to a better robustness to initialization and tracking.

Index Terms— RGB-D SLAM, ORB-SLAM, Visual Odometry, Map Initialization, Triangulation

1. INTRODUCTION

Recent years, Visual SLAM applications have been developed in robots, autonomous cars/drones, AR/VR areas. Even if it only relies on monocular device, the pose estimation Monocular SLAM errors remain more accurate than RGB-D SLAM ones mainly because only a few feature points can be kept in RGB-D SLAM compared with Monocular SLAM. However, Monocular SLAM has a scale ambiguity problem. It is a challenging issue. At a first glance, RGB-D SLAM should be more robust and reliable thanks to the depth information.

Regarding Visual SLAM, the first real-time SLAM using filtering has been proposed by A. J. Davison in 2003 [15]. PTAM (Parallel Tracking and Mapping) [7] using bundle adjustment [16] is limited to small scale environment. ORB-SLAM [1] has been published at the last year improving loop closing, relocalization, and fully automatic initialization based on PTAM. ORB-SLAM achieved better accuracy than the state-of-the-art direct methods, known as LSD-SLAM [8], and RGB-D SLAM [9]. However, it has still ambiguity in difficult scenes and waits a long time to get a reliable initial map (Figure 1). In this paper, we are interested in order to improve ORB-SLAM initialization using depth information. If we use RGB-D cameras which can provide scale information, we can improve this scale ambiguity problem and do not need to wait for initialization step.

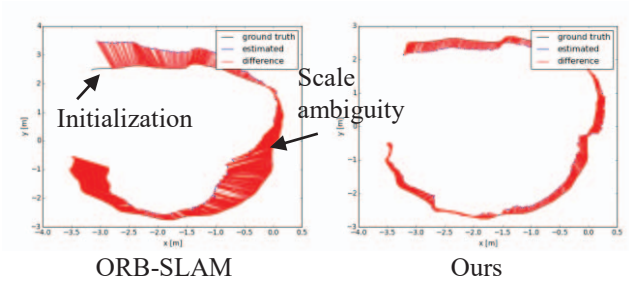


Fig. 1. ORB-SLAM initialization delay and scale ambiguity problem are addressed in this paper.

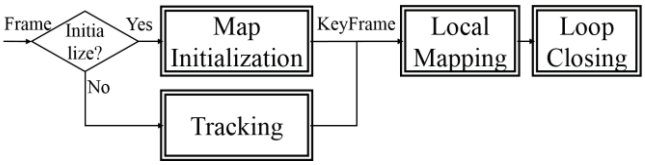


Fig. 2. System overview. The main contents are map initialization, tracking, local mapping, and loop closing.

We propose a novel RGB-D ORB-SLAM which uses not only depth information but also triangulating 3D points out of limited depth area. In our work, depth information gives us initial map. However, RGB-D SLAM has the problem that the depth is gotten in limited area from 0.4 to 8.0m. Depth-less environment such as large room causes tracking lost. To avoid this problem, we use depth information to calculate initial transformation. After that, the system triangulates 3D points in out of limited depth area. We use this new 3D points as initial map. Our method is evaluated using TUM RGB-D Dataset [5]. Our main contributions are following:

1) Solve the scale ambiguity problem.
2) Avoid the initialization delay.
3) Improve the tracking by adding depth to points without depth measurement.

4) Improve the robustness to pure rotations and low texture environments.

2. SYSTEM OVERVIEW

In this section, we explain the differences from ORB-SLAM and our method in 3 parts, map initialization, new keyframe decision, and new points creation. Figure 2 shows an overview of system. The system consists of map initialization, tracking, local mapping, and loop closing.

2.1. Map Initialization

Map initialization goal is to compute the relative pose between reference frame and current frame to triangulate initial map points. This process should not require human selecting a good views pair having significant parallax.

In ORB-SLAM, the system uses homography for planar scenes and fundamental matrix estimation for non-planar scenes to triangulate new initial map points. The system is able to initialize automatically to avoid scale ambiguity. However, we don't need to use this function because we have depth information. Firstly, our initialization system gets reference frame and extracts N (usually N=2000) ORB features. Our method tracks these feature points. Using all of matching points with depth, initial pose is estimated by LM (Levenberg Marquardt) method implemented in g2o [6]. We tried to use Singular Value Decomposition with 3D to 3D point correspondences [10] [11]. However, LM method with 3D to 2D point correspondences can optimize faster and is more accurate. 3D points from depth $\mathbf{X} \in R^3$ and initial pose $\mathbf{T} \in SE(3)$ optimized minimizing the reprojection error with respect to the matched keypoints $\mathbf{x} \in R^2$.

The reprojection error function is:

$$\mathrm{e} = \mathbf{x} - \pi(\mathbf{T}, \mathbf{X}) \tag{1}$$

Where π is the projection function:

$$\pi(\mathbf{T}, \mathbf{X}) = \begin{bmatrix} f_u \dfrac{x}{z} + c_u \\ f_v \dfrac{y}{z} + c_v \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} x & y & z \end{bmatrix}^T = \mathbf{R}\mathbf{X} + \mathbf{t} \tag{3}$$

Where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in R^3$ are respectively the rotation and translation parts of $\mathbf{T}$, (f_u, f_v) and (c_u, c_v) are the focal lengths and the coordinates of the camera center in the standard pinhole camera model [12][13]. The cost function to be minimized is:

$$\mathrm{C} = \sum \rho_{\mathrm{h}} \left(\mathrm{e}^T \Omega^{-1} \mathrm{e} \right) \tag{4}$$

Where ρ_{h} is the Huber robust cost function and $\Omega = \sigma^2 \mathbf{I}_{2\times 2}$ is the covariance matrix. After that, if we have wide baseline, we triangulate points that are out of the limited depth area, using linear triangulation method [14]. We write an inverse projection function:

$$\pi^{-1}(\mathbf{x}, w)^T = w \begin{bmatrix} \dfrac{u - c_u}{f_x} & \dfrac{v - c_v}{f_y} & 1 \end{bmatrix}^T \tag{5}$$

Denoting by $\mathbf{T}(i)^T$ the i-th row of the camera pose $\mathbf{T}$:

$$w \frac{u - c_u}{f_x} = \mathbf{T}(1)^T \mathbf{b}, \; w \frac{v - c_v}{f_y} = \mathbf{T}(2)^T \mathbf{b}, \; w = \mathbf{T}(3)^T \mathbf{b} \tag{6}$$

Eliminating w,

$$\frac{u - c_u}{f_x} \mathbf{T}(3)^T \mathbf{b} = \mathbf{T}(1)^T \mathbf{b}, \; \frac{v - c_v}{f_y} \mathbf{T}(3)^T \mathbf{b} = \mathbf{T}(2)^T \mathbf{b} \tag{7}$$

From initial and current frame, we obtain a total of 4 linear equations in the coordinates of the $\mathbf{b}$, which is written in the form $\mathbf{A}\mathbf{b} = \mathbf{0}$ for a suitable 4×4 matrix, $\mathbf{A}$. This is a non-zero solution for $\mathbf{b}$ solved using the Singular Value Decomposition.

2.2. New KeyFrame Decision

After the initial map is estimated, tracking points allows to estimate the transformation using Local Bundle Adjustment. When the system inserts a new keyframe, the following conditions are required in ORB-SLAM:

1) More than 20 frames have passed.
2) Current frame tracks less than 90% points than reference map points and at least 50 points.

We changed these conditions to improve the robustness of the tracking (especially when fast camera movements occur).

1) Sum of absolute camera translation vector (x, y, and z) is greater than 15cm
2) Sum of absolute camera rotation vector (roll, pitch, and yaw) is greater than 5 degree.
3) Current frame tracks at least 80 points.

2.3. New Map Points Creation

After new keyframe decision, the system triangulates new 3D map points between neighbor keyframes. The system check whether it can be used for new map points using following algorithm.

1) If matching points has Depth in both keyframes, Add new 3D map points from the depth
2) Check parallax between rays
3) Linear triangulation method
4) Check triangulation points are in front of the camera.
5) Check reprojection error in both keyframes
6) Check scale consistency
7) Add new triangulating 3D map points

3. EXPERIMENTAL RESULTS

Our RGB-D ORB-SLAM system is evaluated using TUM RGB-D Dataset [5]. Figure 3 shows absolute trajectory error RMSE (Root Mean Square Error) compared with ORB-SLAM unscaled, scaled by similarity transformation, and our system. In the first and second row, the results of trajectory are similar. In the third row, our system shows a better trajectory than ORB-SLAM.

Experimental environment is windows 8.1, core i7, memory 8GB, C++ implementation. Table 1 shows average processing time in a frame including all process, tracking, mapping, and visualization. Using our approach, the initialization processing time is decreased, because no need to calculate homography and fundamental matrix. To create new map points, local mapping processing time is increased to get more efficient 3D points from depth and triangulate points. Track local map and bundle adjustment times depend on the number of map points.

3.1. Map Initialization

ORB-SLAM has to wait to get a very good initial map to avoid scale ambiguity. In *Fr1/floor* about first 84 frames, only floor is visible in Figure 4(a). The ORB-SLAM system refuses this plane to avoid wrong initialization. The system has to wait till the camera captured a chair in Figure 4(b). Our system can use this plane for initial map.

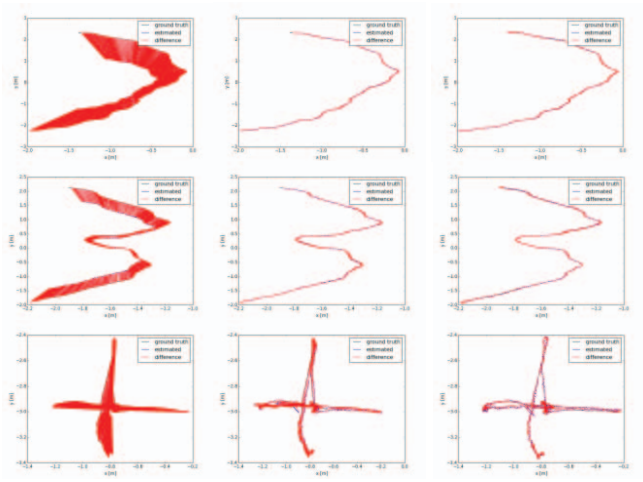
Moreover, Table 2 shows 9 scenes had wrong initialization in ORB-SLAM. It takes average about 200 frames to wait for getting the best initial map. In *Fr3/sitting_static*, *Fr3/walking static*, the ORB-SLAM system never initialize because the camera is static. And in *Fr3/nstr_tex_far*, *Fr3/str_ntex_far*, *Fr3/str_ntex_near* also never initialize because it couldn't get a good parallax in the planar scene.

3.2. Tracking losses

Figure 5 shows example of tracking failure cases. In case of tracking loss scenes, ORB-SLAM doesn't work if camera movement is fast (especially with rotations) and in texture-less environments. Our system takes into account not only depth but also triangulating points in out of limited depth area, and new keyframe selection by threshold of constant movement. Figure 6 shows our system can get more points than ORB-SLAM. It looks much more robust to these motions and various environment. Table 3 shows number of success tracking frames in the analyzed scenes. Our system improved drastically the number of tracking frames.

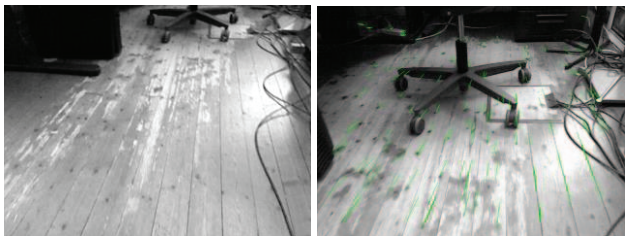
3.3. Accuracy and Robustness

The absolute trajectory error RMSE is shown in Table 4. ORB-SLAM needs a scale of ground truth. There are different scales in almost scenes. In *F3/cabinet*, *Fr3/large_cabinet*, in these scenes are observed scale ambiguity. The absolute trajectory error is not improved well in spite of the trajectory is scaled by similarity transformation. In *Fr3/long_office_household*, ORB-SLAM has just scale unknown, it is not scale ambiguity. Our system recovered scale of ground truth and it is more accurate than ORB-SLAM in 12 scenes. Five scenes are ORB-SLAM could not estimate the trajectory but ours could. In contrary, *Fr3/walking_halfsphere*, there are moving people. In our method, it takes much more feature points than ORB-SLAM, this involves that the accuracy is lower than ORB-SLAM.



ORB (6DOF) ORB (7DOF) Ours (6DOF)

Fig. 3. RMSE (Root Mean Square Error) absolute trajectory error from top, *Fr3/str_tex_far*, *Fr3/str_tex_near*, *Fr3/sitting_ xyz*. Black line is ground truth. Blue line is estimated trajectory. Red line is difference between ground truth and estimated trajectory. Left column is ORB-SLAM unscaled. Middle column is ORB-SLAM scaled using similarity transformation. Right column is our system.



(a) (b)

Fig. 4. Initialization failure frame (a) and success frame (b)

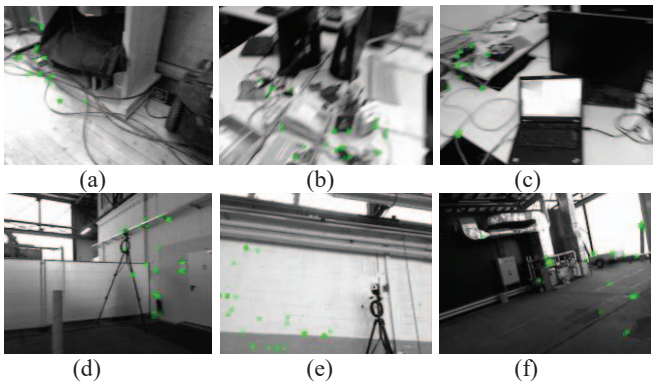


Fig. 5. ORB-SLAM tracking failure cases in (a)*Fr1/floor*, (b)*Fr1/desk2*, (c)*Fr1/ room*, (d)*Fr2/360_kidnap*, (e)*Fr2/large_no_loop*, (f)*Fr2/large_with_loop*, which are fast motion with rotations, blur, texture-less, large, and dark environment.

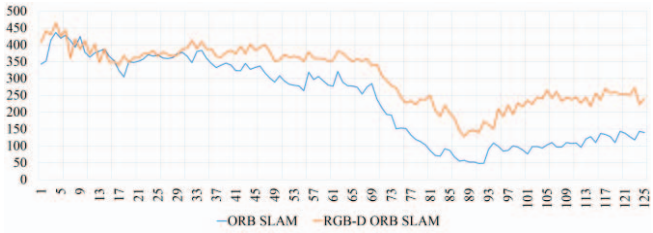


Fig. 6. Example of transition of number of tracking points in *Fr1/floor*. Number of tracking points is in the y-axis and number of frames is in the x-axis. Ours can get more points (depth and triangulating points) than ORB-SLAM. Our method can provide efficient tracking in large environments.

Table 1. Processing time in average including all process, tracking, mapping, and visualization in *Fr1/xyz*.

	ORB-SLAM	Ours
Processing time	279 ms	321 ms

4. CONCLUSION

We proposed a RGB-D ORB-SLAM method improving initialization step using depth information and that is able to compute accurate 3D triangulating points without depth. Our system solves efficiently the scale ambiguity problem and provides also an efficient tracking. Our experiments show clearly how monocular slam has scale ambiguity and tracking failure cases. ORB-SLAM can refuses wrong initialization and consequently, the system waits for a long time to get a great initial map. In contrary our method doesn't need to wait because it already has scale. Using depth, our system can detect and track more efficient 3D points. Depth and triangulation combined is very robust to camera movement and various environment. In the future works, we will improve the tracking process to make it more accurate and more robust in difficult camera movement and scenes.

Table 2. Initialization in ORB-SLAM. Number of success frames and wrong frames. Ours is no failure cases in initialization.

	ORB-SLAM	
	Success	Wrong frames
Fr1/xyz	13	9
Fr1/rpy	71	3
Fr1/desk	54	49
Fr2/rpy	1096	911, 953, 997, 1056
Fr2/desk_person	126	101, 114
Fr3/cabinet	323	48
Fr3/large_cabinet	36	26
Fr3/sitting_rpy	228	40, 198
Fr3/walking_xyz	114	7, 98, 105

Table 3. Comparisons of tracking process with number of tracking frames.

	ORB-SLAM	Ours	
	Tracking	Tracking	ATE [cm]
Fr1/floor	194	1072	13.2285
Fr1/360	24	730	34.5193
Fr1/desk2	156	605	21.3989
Fr1/room	68	1036	23.8968
Fr2/360_hemisph	67	829	10.0468
Fr3/cabinet	341	1109	22.3737
Fr3/large_cabinet	948	976	19.0572
Fr3/sitting_rpy	537	790	24.8820

Table 4. Comparison of absolute trajectory error RMSE. Our system is more accurate than ORB-SLAM in 12 scenes.

	ORB-SLAM		Ours
	Scale	ATE [cm]	ATE [cm]
Fr1/xyz	1.13	1.9963	**1.0086**
Fr1/rpy	1.00	2.6852	4.9636
Fr1/desk	1.13	2.7962	4.0030
Fr2/xyz	1.06	6.6017	**1.5939**
Fr2/desk_person	1.69	3.4228	7.7712
Fr3/long_office	2.36	8.2012	**1.3073**
Fr3/nstr_tex_far	X	X	**3.6973**
Fr3/nstr_tex_near	1.30	2.2904	2.8032
Fr3/str_ntex_far	X	X	**2.7650**
Fr3/str_ntex_near	X	X	**10.6821**
Fr3/str_tex_far	1.95	1.3697	1.7089
Fr3/str_tex_near	1.30	1.7836	**1.4332**
Fr3/sitting_static	X	X	**1.2559**
Fr3/sitting_xyz	6.31	4.7536	**1.6525**
Fr3/walking_halfsph	7.47	4.9457	10.7126
Fr3/walking_rpy	5.45	9.1243	**4.4288**
Fr3/walking_static	X	X	**2.2272**
Fr3/walking_xyz	14.2	4.3395	**2.7162**

5. REFERENCES

[1] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, October 2015.

[2] Raúl Mur-Artal and Juan D. Tardós. "Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM," Robotics: Science and Systems. Rome, Italy, July 2015.

[3] Raúl Mur-Artal and Juan D. Tardós. "ORB-SLAM: Tracking and Mapping Recognizable Features," Robotics: Science and Systems Workshop on Multi VIew Geometry in RObotics, Berkeley, USA, July 2014.

[4] Raúl Mur-Artal and Juan D. Tardós. "Fast Relocalisation and Loop Closing in Keyframe-Based SLAM," IEEE International Conference on Robotics and Automation, Hong Kong, June 2014.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," Proc. of the International Conference on Intelligent Robot Systems, October 2012.

[6] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 3607–3613, May 2011.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, pp. 225–234, November 2007.

[8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," European Conference on Computer Vision, Zurich, Switzerland, pp. 834–849, September 2014.

[9] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," IEEE Transactions on Robotics, vol. 30, no. 1, pp. 177–187, February 2014.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: PAMI-9, Issue: 5, pp. 698-700, September 1987.

[11] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 13, Issue: 4, pp. 376-380, April 1991.

[12] C. Kerl, J. Sturm, and D. Cremers, "Dense Visual SLAM for RGB-D Cameras," International Conference on Intelligent Robot Systems, 2013.

[13] Cha Zhang and Zhengyou Zhang, "Calibration between depth and color sensors for commodity depth cameras," IEEE International Conference on Multimedia and Expo, Barcelona, July 2011.

[14] Richard I. Hartley and Peter Sturm,"Triangulation," Computer Vision and Image Understanding, Vol. 68, No. 2, pp. 146-157, November 1997.

[15] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," IEEE International Conference on Computer Vision, vol. 2, pp.1403-1410, 2003.

[16] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real-time localization and 3d reconstruction," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 363-370, 2006.