Dense 3D Mapping Using Volume Registration

Luke Lincoln(✉) and Ruben Gonzalez

Information and Communication Technology, Griffith University,
Gold Coast, Australia
{l.lincoln,r.gonzalez}@griffithuni.edu.au

Abstract. In this paper, a novel closed form solution is presented for solving the simultaneous localization and mapping (SLAM) problem. Unlike existing methods which rely on iterative feature matching, the proposed method utilises 3D phase correlation. This method provides high noise robustness, even in the presence of moving objects within the scene which are problematic for SLAM systems. Quantitative and qualitative experimental results are presented, evaluating the noise sensitivity, reconstruction quality and robustness in the context of moving objects.

1 Introduction

Simultaneous localization and mapping (SLAM) has many applications in robotics, architecture and engineering, business and science. Its objective is to produce a map (2D birds-eye-view, or 3D) of an environment using image and other sensory data. This is typically performed by computing local features, iteratively matching them across frames and solving for the camera pose and location. This feature matching approach is dependent on finding a sufficient number of matches. When this is true the approach is able to cope with local matching disparities by treating them as outliers. This technique is not robust when features are not stable or when feature confusion occurs.

1.1 Monocular Camera Feature Based Systems

Monocular Feature based SLAM systems use feature matches to estimate camera pose and location changes across frames [4]. Variations of this method use different features including: corners and lines [13], image patches [24] and exemplar feature matching [3]. SIFT features are used most often in SLAM [1,7,12,21], in addition FAST features have been explored [15–18]. Beall et al. [1] made use of both SIFT and SURF features in their underwater SLAM system. Real-time monocular SLAM systems based on this approach have also been proposed [3,21]. RANSAC is often used in monocular SLAM [7,15–17,23] to remove outliers which cause incorrect camera parameter estimates. Bundle adjustment is also used as an additional step to refine camera parameter estimation [7].

P.C. Vinh et al. (Eds.): ICTCC 2016, LNICST 168, pp. 22–32, 2016.
DOI: 10.1007/978-3-319-46909-6_3

1.2 Stereo Camera Feature Based Systems

Stereo based SLAM systems also use features to estimate camera parameters. However, stereo based systems are capable of generating dense depth data more easily using stereo algorithms. Miro et al. [19] proposed a stereo based method which uses SIFT and the extended Kalman filter. The method by Van Gool et al. [22] works with un-calibrated stereo pairs. It uses Harris corner features and a multi-view stereo algorithm. Sim et al. [25] and Gil et al. [8] both presented stereo based SLAM systems which use SIFT.

1.3 RGB-D Sensor Feature Based Systems

RGB-D SLAM systems use both depth and image data and are capable of generating dense 3D reconstructions. Many of these methods rely on feature matching techniques [5,6,10]. RANSAC is often used to filter outliers for the estimation of camera parameters [5,6,10]. Another method which has also been used extensively in the area is Iterative Closest Point (ICP) [2,6,10,11,20,26]. ICP iteratively registers point cloud data, and is used to refine camera parameter estimates. A method named KinectFusion was proposed by Newcombe et al. [20] which uses RANSAC and a GPU implementation of IPC. Whelan et al. [28] extended this method allowing it to map larger areas using Fast Odometry From Vision (FOVIS) over ICP. Bylow et al. [2] improved the ICP approach by registering data using a signed distance function.

1.4 Non-Feature Based Methods

Several RGB-D SLAM systems are also non-feature based [11,14,27]. Weikersdorfer et al. [27] presented a novel sensor system named D-eDVS along with an event based SLAM algorithm. The D-eDVS sensor combines depth and event driven contrast detection. Rather than using features, it uses all detected data for registration. Kerl et al. [14] proposed a dense RGB-D SLAM system which uses a probabilistic camera parameter estimation procedure. It uses the entire image rather than features to perform SLAM.

1.5 Summary

As is evident from the current literature, SLAM typically relies on feature matching and RANSAC. However, these approaches fail when there are too few features, when feature confusion occurs or, when features are non-stationary due to object motion. As the extent of random feature displacement becomes more global the effectiveness of these approaches diminishes. Feature matching also dominates in image registration. However, Fourier based methods have been shown to work well under larger rotations and scales [9] whilst being closed form, insensitive to object motion and scaling naturally to GPU implementations. Accordingly, we propose a novel, closed form Fourier based SLAM method.

2 Method

The proposed SLAM method consists of various steps. First each frame f_i that is captured, consisting of a colour and depth image pair is projected into 3D space, forming colour point cloud $points_i$ and re-sampled into a volume V_i. Then, the transform parameters between pairs of volumes V_i and V_{i+1} are estimated using $VolumeRegister_{\theta\varphi t_x t_y t_z}$ shortened to $VR_{\theta\varphi t_x t_y t_z}$. These parameters are used to update transformation matrix M. The points corresponding to f_2 ($points_1$) are then transformed using the updated M matrix and added to the cumulative $PointCloud$ database. Two lists, $Cameras$ and $Poses$, are also updated to track camera pose and location per frame. This basic procedure is given in Listing 1 and elaborated upon in subsequent subsections.

```
f1 = ReadFrame();
PointCloud = project(f1);
M = IdentityMatrix(), Camera = [0,0,0]^T, Pose = [0,0,1]^T;
Cameras = [Camera], Poses = [Pose];
while(more frames){
    f2 = ReadFrame();
    points1 = project(f2), points2 = project(f1);
    V1 = ResampleVolume(points1), V2 = ResampleVolume(points2);
    (θ, φ, tx, ty, tz) = VRθφtxtytz(V1, V2);
    M = M× TransformMatrix((θ, φ, tx, ty, tz));
    points1 = Transform(points1, M);
    PointCloud = PointCloud ∪ points1;
    Camera = M^-1 × Camera;
    Pose = M^-1 × Pose;
    Cameras.add(Camera);
    Poses.add( (Pose−Camera) / |Pose−Camera| );
    f1 = f2;
}
```

Listing 1. Phase Correlation Based SLAM Algorithm

2.1 Sensor Input

The input to our method is a color and depth image pair, $f(u, v)$ and $g(u, v)$ obtained using an Asus Xtion PRO LIVE sensor at a resolution of 640×480. Each pixel is projected into 3D space using $X_{u,v} = \frac{(u-c_x)Z_{u,v}}{f}$, $Y_{u,v} = \frac{(v-c_y)Z_{u,v}}{f}$ and $Z_{u,v} = g(u, v)$. Here, $[c_x c_y]^T$ represent the center of the image whilst f represents the focal length, defined as 525.0. The point clouds generated by projecting these images are then quantized into image volumes. Results reported in this paper were obtained using volumes of 384^3 voxels in size.

2.2 Volume Registration

Figure 1 shows a functional block diagram of our method. The input data are two 3D volumes ($Volume_1$ and $Volume_2$) and the output is the transformation

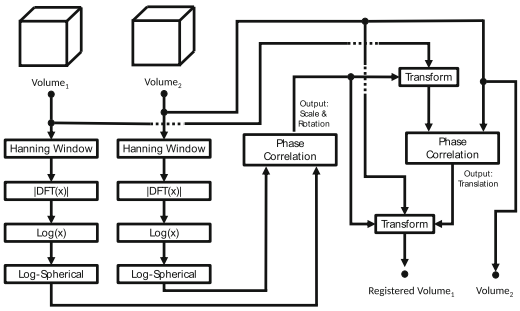


Fig. 1. System diagram for registration process

matrix required to register the two volumes. The volumes are first Hanning windowed. Next, a translation independent representation is obtained for each by taking the magnitude of their 3D FFTs. Then a log function is applied to the resulting magnitude values, improving scale and rotation estimation [9]. Following a log-spherical transformation, 3D phase correlation is performed to find the global rotation and scale relationship between $Volume_1$ and $Volume_2$. $Volume_1$ is then inversely transformed by the rotation and scale parameters, leaving only the translation to be resolved. This is found by applying phase correlation again between the transformed $Volume_1$ and $Volume_2$.

2.3 Phase Correlation

Given a volume V_1 and a spatially shifted version of it V_2, the offset can be recovered using $PhaseCorrelation$ (Eq. 1). This function takes two volumes as input and returns the translation between them.

$$(x, y, z) = PhaseCorrelation(V_m, V_n) \tag{1}$$

The $PhaseCorrelation$ function first applies 3D FFTs to volumes, V_1 and V_2, converting them into the frequency domain, i.e. $F_{1_{x,y,z}} = FFT(V_1)$ and $F_{2_{x,y,z}} = FFT(V_2)$. Taking the normalised cross power spectrum using Eq. 2 completes the Phase correlation function.

$$F_{3_{x,y,z}} = \frac{F_{1_{x,y,z}} \circ F^*_{2_{x,y,z}}}{|F_{1_{x,y,z}} \circ F^*_{2_{x,y,z}}|} \tag{2}$$

Here, $\circ$ is an element-wise multiplication and $|x|$ is the magnitude function. Taking the inverse FFT of F_3, gives the phase correlation volume V_3 ($V_3 = FFT^{-1}(F_3)$). The location of the peak value in V_3, (x_1, y_1, z_1) gives the shift between the V_1 and V_2. The phase correlation volume is typically noisy making the peak difficult to locate.

2.4 Recovering Scale, Rotation and Translation Parameters

If V_1 and V_2 are instead rotated and scaled versions of the same volume, such that they are related by some translation (t_x, t_y, t_z), y-axis rotation θ, and scale φ. Further action is required to recover translation parameters. The first step, given two volumes V_1 and V_2 of size N^3 is to apply a Hanning windowing function (Eq. 3).

$$HW_{x,y,z}=\frac{1}{2}\left(1-cos\left(\frac{2\pi\left(\sqrt{\left(\frac{N}{2}\right)^3}-\sqrt{\left(x-\frac{N}{2}\right)^2+\left(y-\frac{N}{2}\right)^2+\left(z-\frac{N}{2}\right)^2}\right)}{2\sqrt{\left(\frac{N}{2}\right)^3}-1}\right)\right) \tag{3}$$

The rotation and scale factors are recovered first using a translation independent representation of the volumes using the Fourier shift theory. For this, the magnitude of the FFT of the volumes is taken, $M_1 = |FFT(V_1)|$, $M_2 = |FFT(V_2)|$. The zero-frequency of both M_1 and M_2 is shifted to the center of the volume and the log of the result is taken $M_1' = Log(M_1)$, $M_2' = Log(M_2)$ which reduces noise on the phase correlation volume. A log-spherical transform is then used to turn rotation and scaling into translation for both M_1' and M_2'. Equation 4 shows the corresponding log-spherical space coordinate $(X_{log-spherical}, Y_{log-spherical}, Z_{log-spherical})$ for a given (x, y, z) euclidean space coordinate.

$$\begin{aligned} X_{log-spherical} &= \frac{atan\left(\left(\frac{x-\frac{N}{2}}{\sqrt{x^2+y^2+z^2}}\right)\left(\frac{y-\frac{N}{2}}{\sqrt{x^2+y^2+z^2}}\right)^{-1}\right)N}{360} \\ Y_{log-spherical} &= \frac{acos\left(\frac{y}{\sqrt{x^2+y^2+z^2}}\right)N}{180} \\ Z_{log-spherical} &= \frac{log\left(\sqrt{x^2+y^2+z^2}\right)N}{log\left(\frac{N}{2.56}\right)} \end{aligned} \tag{4}$$

The log-spherical transforms of M_1' and M_2' are then phase correlated to find the shift between them, $(x_{M'}, y_{M'}, z_{M'}) = PhaseCorrelation(M_1', M_2')$. The rotation θ and scale φ factors between V_1 and V_2 can then be found from the shift parameters using Eq. 5 .

$$\begin{aligned} \theta &= \frac{-360x_{M'}}{N} \\ \varphi &= e^{-\left(2.56^{-1}N\right)z_{M'}N^{-1}} \end{aligned} \tag{5}$$

Using θ and φ, V_1 can now be inverse transformed (using $(\frac{N}{2}, \frac{N}{2}, \frac{N}{2})$ as the origin). This aligns V_1 and V_2 with respect to scale and y-axis rotation. The translation parameters (t_x, t_y, t_z) can then be found using phase correlation as given in Eq. 6.

$$(t_x, t_y, t_z) = PhaseCorrelation(scale(rotate(V_1, \theta), \varphi), V_2) \tag{6}$$

The complete function to recover translation, rotation and scaling, combining Eqs. 2–6 as is denoted in Listing 1 is 7.

$$(\theta, \varphi, t_x, t_y, t_z) = PhaseCorrelation_{\theta\varphi t_x t_y t_z}(V_m, V_n) \tag{7}$$

2.5 Performance Analysis

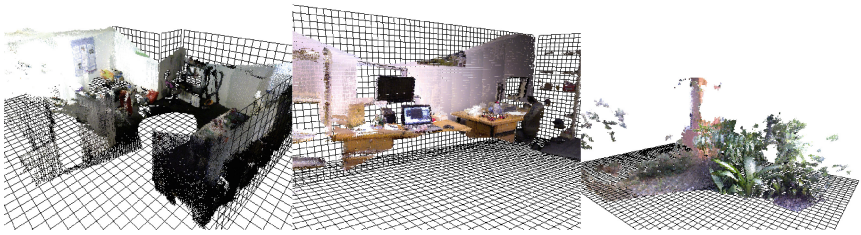
To assess the performance of our method, the size of the volumes being registered is defined as N^3 whilst each frame is sampled at a resolution of $W \times H$. The projection process requires $12WH$ operations whilst re-sampling the point cloud requires $2WH$ operations. The Volume Registration process, $VolumeRegister\theta\varphi t_x t_y t_z(V_1, V_2)$ consists of 2 × Hanning windowing processes, 2 × 3D FFTs, 2 × volume-logs, 2 × log-spherical transforms, 2 × phase correlation processes and 1 × linear transformation and peak finding.

The Hanning windowing function requires 26 operations. The 3D FFT has complexity of $3N^3 \log N$, the log and log-spherical transform functions require 3 and 58 operations per voxel respectively. Multiplying two frequency spectra together and transforming a volume requires 15 and 30 operations per voxel respectively. Finding the peak value requires $2N^3$ operations. The complexity in terms of number of operations for the phase correlation process is given in Eq. 8 This process requires 2 × 3D FFTs, 1 × frequency spectra multiplication, and 1 × peak finding operation.

$$6N^3 \log N + 2N^3 + 15 \tag{8}$$

The total complexity can then be found by taking into account the projection and re-sampling totals as well as the total for $VolumeRegister\theta\varphi t_x t_y t_z(V_1, V_2)$. Tallying the number of operations for each process and multiplying them by number of times the process is performed gives us the number of operations as a function of W, H and N in Eq. 9.

$$6N^3 + 28WH + 18(N^3 \log N) + 230 \tag{9}$$



(a) Apartment (b) Office (c) Garden

Fig. 2. Reconstructed scenes.

3 Experiments

A number of experiments were undertaken to assess the reconstruction accuracy, noise sensitivity and robustness to object motion. Experiments were performed using an ASUS Zenbook UX303LN with an Intel i7 5500u Dual Core 2.4 GHz processor, 8 GB of RAM and an NVIDIA GE-FORCE 840 M GPU. For volumes of 384^3, 1 $\times$ registration per second was possible. To achieve real-time performance, 1 out of every 30th frames was processed.

3.1 Reconstruction Quality

To assess reconstruction accuracy, two indoor environments (Apartment and Office) as well as one outdoor environment (Garden) were used, these can be seen in Figs. 2a, b and c respectively. The Apartment reconstruction was recorded by moving through a room whilst rotating the camera. Some frames contained nothing but featureless walls, others had contrast shifts due to the camera's automatic contrast feature, yet, accurate reconstruction was achieved. The office reconstruction was generated by rotating the camera about the y-axis while moving backwards. Whilst our method is a closed form solution, its accuracy is still comparable to existing feature based SLAM methods. Typical feature based methods work well with indoor environments where local features are readily distinguishable and easy to match. They do not tend to work as well with complex outdoor scenes where feature confusion is likely. To assess performance in such outdoor scenes, a garden scene containing bushes, plants and a ground covering of bark and rocks was used. In the case of a feature matching approach this scene would likely result in feature confusion, making camera tracking difficult. The proposed method was able to produce a good quality reconstruction. Hence, our approach readily overcomes difficulties common to feature matching methods.

3.2 Noise Sensitivity

To assess robustness to noise, the estimated camera parameters are compared to ground truth data under different noise conditions. In each experiment, varying amounts of random noise were added per voxel prior to registration. This is expressed in decibels using the Signal to Noise Ratio (SNR). Each voxel value lies in the range [0–1]. Here, a noise value of 10 % means random noise was added in the range [−0.05, 0.05]. Tracking error is measured in centimetres and voxel error (the error in the phase correlation volume). The first experiment evaluated noise robustness whilst the camera was translated by varying amounts (5 cm, 10 cm and 15 cm). Results in Table 1 show that, for camera translations up to 15 cm and SNR values above 6.0 our method is robust to noise. At video rates, a displacement of 10 cm per frame equates to a camera velocity of 3 m/s (about twice the normal walking speed).

Table 2 shows the results for tracking camera rotations of 10, 20 and 30 degrees per frame. At video rates, 12 degrees per frame is almost a full rotation per second. In rotations of 10 degrees, the error was less than a degree for all

Table 1. Translation tracking

Translation (cm)	Noise range (%)	SNR	Error (cm)	Error (voxel)
5 cm	0	∞	0	0
5 cm	10	20 db	0	0
5 cm	25	12 db	0	0
5 cm	50	6 db	0	0
5 cm	75	2.5 db	112.28	89.83
10 cm	0	∞	0	0
10 cm	10	20 db	0	0
10 cm	25	12 db	0	0
10 cm	50	6 db	156.65	125.32
15 cm	0	∞	2.8	2.24
15 cm	10	20 db	2.8	2.24
15 cm	25	12 db	2.8	2.24
15 cm	50	6 db	198.55	158.84

Table 2. Rotation tracking

Rotation	Noise (%)	SNR	Error (θ)	Error (voxel)
10°	0	∞	0.31	0
10°	10	20 db	0.31	0
10°	25	12 db	0.63	1
10°	30	10.5 db	90.62	96
20°	0	∞	0.31	0
20°	10	20 db	0.63	1
20°	15	16.5 db	38.13	40
30°	0	∞	3.75	4
30°	10	20 db	3.28	3
30°	15	16.5 db	30	32

Table 3. Object motion test

Object size	Error (cm)	Error (voxel)
0.35	0	0
2.95	0	0
6.22	0	0
12.28	0	0
19.82	0	0
22.39	0	0
26.09	0	0
31.00	0	0
48.23	38.42	15
74.32	113.57	44

but a noise level of 30 % and above. This base line error is due to the sampling resolution of the volume, as voxel error was in fact zero. As with pure translation, the effect of noise increases with camera disparity. At 30 degrees, little matching information is available. However, for noise levels of 10 % or less, voxel distance error was as low as 4 with an angular error less than 3.8. Rotations of this magnitude are unlikely, moreover motion blur would occur.

3.3 Robustness to Object Motion

To assess robustness to object motion, experiments were conducted by moving the camera backwards along the z-axis by 5 cm per frame whilst moving objects in and out of the scene so that they only appear in one of the volumes being registered. Various sized objects including stacks of CDs, large boxes, people and several pieces of furniture were used and are measured by the percentage of the frame they occupy. Results from Table 3 show the proposed method was accurate upto an object size of 31 %, but failed for objects taking up over 48.23 %.

4 Conclusion

In this paper, we proposed a novel non-feature based approach to SLAM, which can generate accurate 3D color reconstructions of both indoor and outdoor environments. This method is a closed form solution, scales naturally to the GPU, and is shown to be robust to global noise and object motion. Future work in this area includes investigating a system to recover from misregistered frames.

References

1. Beall, C., Dellaert, F., Mahon, I., Williams, S.B.: Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys. In: 2011 IEEE-Spain OCEANS, pp. 1–6. IEEE (2011)
2. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: Robotics: Science and Systems (RSS) Conference 2013, vol. 9 (2013)
3. Chekhlov, D., Pupilli, M., Mayol, W., Calway, A.: Robust real-time visual slam using scale prediction and exemplar based feature description. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–7. IEEE (2007)
4. Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 865–880 (2002)
5. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the RGB-D slam system. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1691–1696. IEEE (2012)
6. Engelhard, N., Endres, F., Hess, J., Sturm, J., Burgard, W.: Real-time 3d visual slam with a hand-held RGB-D camera. In: Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden, vol. 180 (2011)
7. Eudes, A., Lhuillier, M., Naudet-Collette, S., Dhome, M.: Fast odometry integration in local bundle adjustment-based visual slam. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 290–293. IEEE (2010)
8. Gil, A., Reinoso, O., Mozos, O.M., Stachniss, C., Burgard, W.: Improving data association in vision-based slam. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2076–2081. IEEE (2006)
9. Gonzalez, R.: Improving phase correlation for image registration. In: Image and Vision Computing New Zealand 2011 (IVCNZ 2011) (2011). http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1002602

10. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using depth cameras for dense 3d modeling of indoor environments. In: The 12th International Symposium on Experimental Robotics (ISER). Citeseer (2010)
11. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 559–568. ACM (2011)
12. Jensfelt, P., Kragic, D., Folkesson, J., Bjorkman, M.: A framework for vision based bearing only 3d slam. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 1944–1950. IEEE (2006)
13. Jeong, W.Y., Lee, K.M.: Visual slam with line and corner features. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2570–2575. IEEE (2006)
14. Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for RGB-D cameras. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2100–2106. IEEE (2013)
15. Konolige, K., Agrawal, M.: Frameslam: From bundle adjustment to real-time visual mapping. IEEE Trans. Robot. **24**(5), 1066–1077 (2008)
16. Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., Fua, P.: View-based maps. Int. J. Robot. Res. **29**(8), 941–957 (2010)
17. Kundu, A., Krishna, K.M., Jawahar, C.: Realtime motion segmentation based multibody visual slam. In: Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, pp. 251–258. ACM (2010)
18. Leelasawassuk, T., Mayol-Cuevas, W.W.: 3d from looking: using wearable gaze tracking for hands-free and feedback-free object modelling. In: Proceedings of the 17th Annual International Symposium on Wearable Computers, pp. 105–112. ACM (2013)
19. Miro, J.V., Zhou, W., Dissanayake, G.: Towards vision based navigation in large indoor environments. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2096–2102. IEEE (2006)
20. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 127–136. IEEE (2011)
21. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., et al.: Detailed real-time urban 3d reconstruction from video. Int. J. Comput. Vis. **78**(2–3), 143–167 (2008)
22. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comput. Vis. **59**(3), 207–232 (2004)
23. Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., Bathiche, S.: Monofusion: real-time 3d reconstruction of small scenes with a single web camera. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 83–88. IEEE (2013)
24. Silveira, G., Malis, E., Rives, P.: An efficient direct approach to visual slam. IEEE Trans. Robot. **24**(5), 969–979 (2008)
25. Sim, R., Elinas, P., Griffin, M., Little, J.J., et al.: Vision-based slam using the rao-blackwellised particle filter. In: IJCAI Workshop on Reasoning with Uncertainty in Robotics, vol. 14, pp. 9–16 (2005)

26. Stückler, J., Behnke, S.: Robust real-time registration of RGB-D images using multi-resolution surfel representations. In: 7th German Conference on Robotics; Proceedings of ROBOTIK 2012, pp. 1–4. VDE (2012)
27. Weikersdorfer, D., Adrian, D.B., Cremers, D., Conradt, J.: Event-based 3d slam with a depth-augmented dynamic vision sensor. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 359–364. IEEE (2014)
28. Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., McDonald, J.: Kintinuous: spatially extended kinectfusion (2012)