Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

3D Reconstruction of Dense Model based on the Sparse Frames using RGBD Camera

Wenbo Han[1], Xiaomeng Liu[1], Shuang Song[1]*, Max Q.-H. Meng[2], *Fellow, IEEE*

***Abstract*—With the popularity of consumer-grade depth cameras on mobile devices, 3D reconstruction based on the RGBD camera has once again become a hot topic in the field of 3D vision. The current application of RGBD cameras is mainly in the field of large-scale 3D reconstruction and VSLAM that rely on high-performance graphics cards to achieve real-time reconstruction and currently reach a relatively mature stage. The reconstruction of specific scene models (such as objects, human bodies, human faces) has a lot of application prospects in the fields of intelligent volume measurement, face recognition, motion capture, etc, attracting the attention of researchers. In this paper, a lightweight 3D reconstruction algorithm framework is designed for scene objects. The selecting frame module is introduced. The dense 3D reconstruction based on sparse frames is implemented on the Raspberry Pi 4B to improve the model reconstruction efficiency and we design the point cloud pose-processing module to improve the quality of the model reconstruction.**

***Index Terms*—3D reconstruction, RGBD, scene model, select frame, pointCloud post-processing.**

I. Introduction

With the development of hardware level, more and more mobile devices are implanted with high-quality depth sensors and people's needs are no longer satisfied with two-dimensional planes. Research in the field of three-dimensional vision has once again become a hot spot. The research and application of lightweight model reconstruction algorithms are also current difficulties. This paper uses the Raspberry Pi 4B and Real SenseD435 as own hardware platform to build own 3D reconstruction algorithm framework, verify the lightweight of the algorithm and realize its own rapid modeling for scene objects.

3D reconstruction research based on RGBD Camera is mainly applied to large scenes such as outdoor and indoor to apply to the unmanned driving and robot navigation positioning [1]. Currently, the 3D reconstruction of the objects mainly depends on 3D scanning equipment. The reconstructed model lacks texture features and the reconstruction cost is high. The scanning device is bulky and difficult to carry. It is far from the popular application of 3D technology. The reconstruction of the scene model based on the RGBD camera has many problems in pose estimation, reconstruction quality and efficiency.

Acquisition of camera pose: Because the texture of the object and the surrounding background are not rich, it is difficult to solve the pose using the feature point method (such as ORB [2], [3], SIFT [4], SURF [5], etc) and its combination method (such as imu+camera [6], feature point + ICP [7], imu+ICP [8], etc) is not reliable here. The amount of calculation of using the ICP(Iterrative Closest Point) [9] to solve pose is large and the real-time reconstruction depends on the high Configuring hardware acceleration so that is difficult to achieve lightweight [10], [11]. Multiple cameras can also be used to build a data acquisition system, but the cost is too high.

Grid model reconstruction quality: The model of the object can be Ideally reconstructed with four images, but the quality of the model is directly affected by the error of the measuring device and the accumulated error of the calculation of each module. In order to compensate for the lack of information, many schemes use dense frame fusion [10]. When more information is incorporated, more noise is added. In addition, the pose correctness of each frame cannot be guaranteed when solving the pose. Based on the dense frame fusion schemes,the reconstruction quality cannot be guaranteed and the reconstruction efficiency is also low.

The issue of reconstruction efficiency: There are many factors affecting the reconstruction efficiency of 3D reconstruction. Except for dense frame fusion, whether to filter the effective point cloud, whether to over-was existing and memory, and whether the algorithm is optimized, etc., all have a large proportion in improving efficiency.

Reconstruction efficiency and reconstruction quality are intuitive feelings that affect the user's experience. How to improve model reconstruction quality and reconstruction efficiency is the focus of this paper. In view of the above problems, this paper proposes a lightweight framework for dense 3D reconstruction based on sparse frames (Fig.1). As follow:

(1) In terms of pose solving, this paper introduces two different sizes of checkerboards as markers. By determining

[1] Wenbo Han, Xiaomeng Liu, Shuang Song are with School of Mechanical Engineering and Automation, Harbin Institute of Technology(Shenzhen), Shenzhen, China, 518055.

[2] Max Q.-H. Meng is with Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong, China, and affiliated with the State Key Laboratory of Robotics and Systems (HIT), Harbin Institute of Technology, China.

*Corresponding author: Shuang Song, mail: songshuang@hit.edu.cn

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

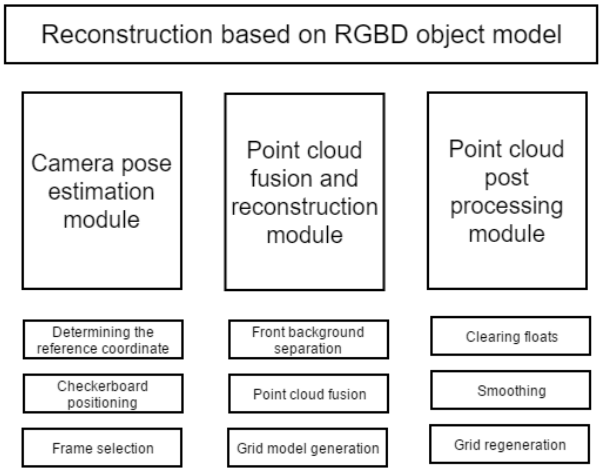
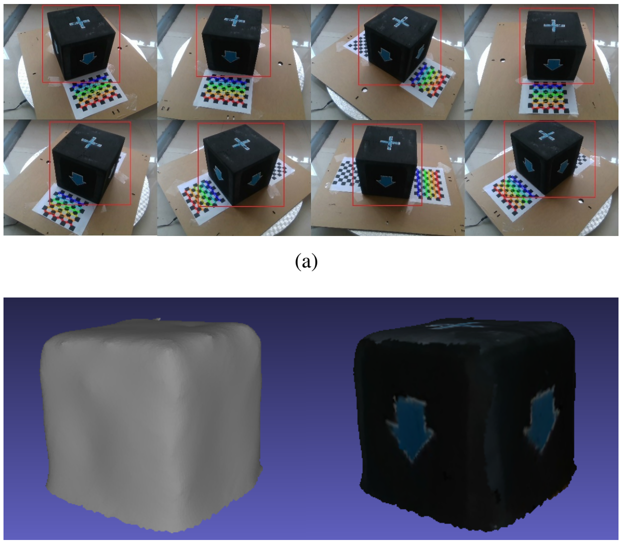

Reconstruction based on RGBD object model
Camera pose estimation module
Point cloud fusion and reconstruction module
Point cloud post processing module
Determining the reference coordinate
Checkerboard positioning
Frame selection
Front background separation
Point cloud fusion
Grid model generation
Clearing floats
Smoothing
Grid regeneration

Fig. 1. Algorithm framework.



(a)

(b)

Fig. 2. (a) 8 RGB images after checkerboard detection, the target detection and filtering frame. (b) Reconstruction result after fusion reconstruction and point cloud post-processing.

the reference coordinate system and the reference 3D points, Pnp [12] is used to settle the pose. By analyzing the motion information of the camera, the incorrect pose is eliminated. Finally, 8 images of different angles are properly selected in the correct pose set.

(2) In the fusion and reconstruction: combining the aligned rgb and depth image to extract the effective point cloud by using the Yolov2 algorithm [13] to predict the rgb image and output the boundingbox of the model to be reconstructed. Using the structure of Voxel Hash [14], the grid model (rough) is produced by point cloud fusion and moving cube method using TSDF [15].

(3) Pointcloud post-processing: the generated mesh model has problems such as floating, hollow, and rough model, so introducing Poisson reconstruction, smoothing processing, de-floating and etc to improve the reconstruction effect of the model.

II. RELATED WORK

Vision-based 3D reconstruction can be divided into two categories: sparse and dense reconstruction. Sparse reconstruction is based on key points, mainly used in the field of VSLAM, such as ORB-SLAM2, VINS-Mono, etc, but the image utilization rate is not high. With the improvement of hardware level and the accumulation of technology, the application of dense 3D reconstruction is more and more extensive. More attention is paid to the construction of dense models in my work.

Vision-based dense model reconstruction can be divided into active vision and passive vision. The difference between the two lies mainly in the way of obtaining depth images. Active vision reconstruction refers to the technique of using the TOF, structured light or infrared stereo vision technology to measure the surface of an object, and then combining color image information to reconstruct the surface model of the object (such as KinectFusion, ElasticFusion [16], etc). Passive visual reconstruction refers to the technique of reconstructing an object model by acquiring image sequences through one or more vision sensors and using the theory of multi-view geometry. According to the number of vision sensors used, it can be divided into monocular, binocular stereo vision and multi-view reconstruction. Passive vision can be applied well in the complex environments. Compared with active vision, the requirements for the environment are not very high and the adaptability is strong. But its method is difficult to process data, relies on computing resources and cannot gain the high reconstruction accuracy.

Since Microsoft introduced the Kinect camera, the research based on the RGBD 3D reconstruction has emerged one after another. KinectFusion is the originator of RGBD based 3D reconstruction. This algorithom realizes real-time indoor scene reconstruction relying on the high-configuration graphics card by using the ICP method to solve the camera pose and using the TSDF to fusion.

Since KinectFusion needs to initialize the size of the volume before rebuilding, it causes existing waste, which affects the reconstruction efficiency. Chisel [14] uses the Voxel Hash-based data structure to solve this problem and the IMU + Camera method to obtain the motion pose and implements the real-time 3D reconstruction of the indoor scene on the Google Tango mobile phone. That is a breakthrough in the application of 3D technology on the mobile end.

For the modeling of specific scene models, FaceWarehouse [17] adopts the KinectFusion method and use multiple Kinect cameras to build a data acquisition system to reconstruct the face model. There is so obvious depth difference between the frontgrounds and backgrounds that it achieves background separation well according to the depth value.

For the reconstruction of scene objects, there are high requirements for reconstruction quality and reconstruction efficiency. At present, there is no relevant technical application and open source.

The purpose of our work is to apply the 3D reconstruction algorithm to the mobile platform, which contributes to the intelligent volume measurement, arguement/visual reality, etc. A lightweight framework of object model reconstruction algorithm is proposed (Fig.3).

III. SYSTEM IMPLEMENTATION

A. Pose estimation

1) Determine the reference coordinates and reference 3D points: when collecting images, it ensures that the first frame contains two marks and the z-axis direction is as perpendicular to the ground as possible. The second frame and the third frame only contain marker1 and marker2 respectively. The 3D points of marker1 and marker2 are extracted from the second frame and the third frame respectively and using the Pnp method by remapping them to the reference coordinate system obtains the reference 3D point cloud coordinates.

Although the foreground of the RGBD cameras is infinite, there are still many problems in the depth map output of the depth camera limited by the physical hardware, such as smooth objects, translucent objects, surface reflection of dark objects, out of range, etc. For the chessboard corners without depth value, the Breadth-First-Search method is used to obtain the substitution value (Fig.4).

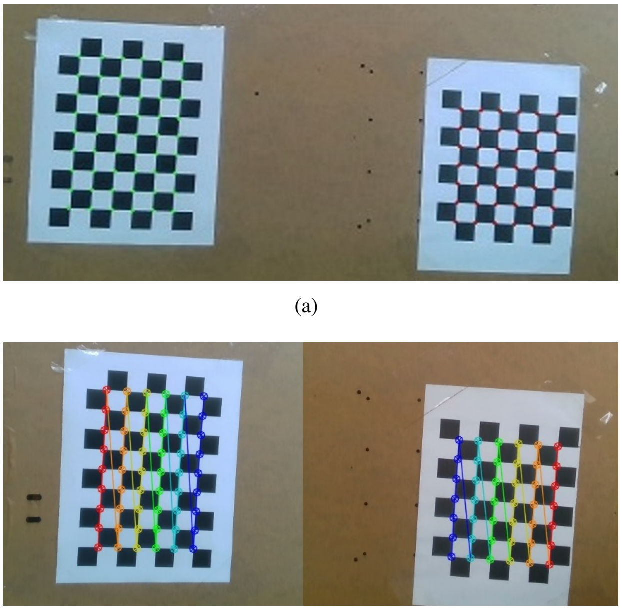
2) Camera motion state analysis: in order to improve the quality and efficiency of model reconstruction, the motion state is analyzed according to the camera pose and the image with correct pose is selected to implement the 3D reconstruction based on the sparse frames. R is the rotation matrix given by the definition of Z-Y-X Euler angle ($c\alpha$ means $\cos\alpha$, $s\alpha$ means $\sin\alpha$).

$$R = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\alpha s\beta & c\beta c\gamma \end{bmatrix} \quad (1)$$

α, β and γ are independent of each other, corresponding to the rotation angle of the Z-axis, Y-axis and X-axis. Ideally, when collecting data, we fixed the camera and rotated the dial to obtain an image of the object with multiple angles of view (approximating the Z-axis around the reference coordinates). Therefore, the trend of α is approximately linear over the range [0, 2π] (Fig.5). And the values of roll and pitch should be approximately constant.

$$R_{21} = s\alpha c\beta \quad (2)$$

$$R_{23} = s\alpha s\beta c\gamma - c\alpha s\gamma \quad (3)$$



(a)

(b)

Fig. 4. (a) Remapping from the second and third frame to first frame. (b) The Corners detection of the second and third frame.

The curve of R_{21} and R_{23} should satisfy the trend of sinusoidal function. According to this correspondence, we can intuitively see which frame pose results are wrong. Combined with the changing trends of yaw, pitch and roll, we can eliminate the error frame.

When collecting data, we fixed the camera and rotated the dial to obtain an image of the object with multiple angles of view (approximating the Z-axis around the reference coordinates). Therefore, the trend of α is approximately linear over the range [0, 2π] (Fig.5).

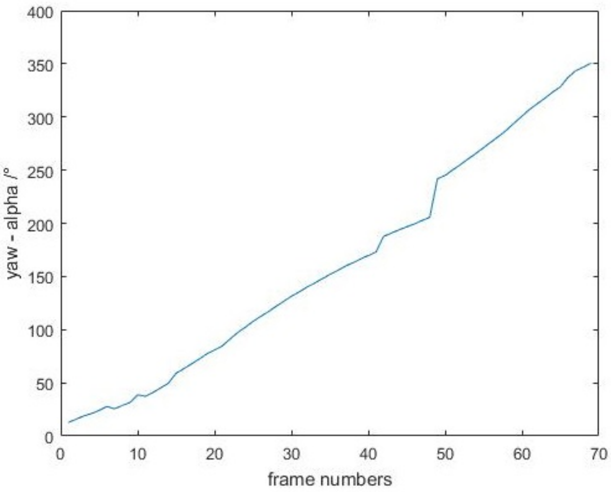


Fig. 5. Yaw angle diagram.

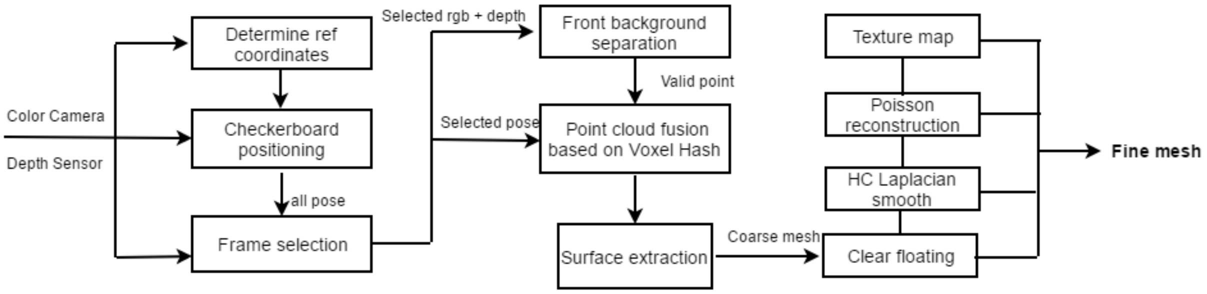
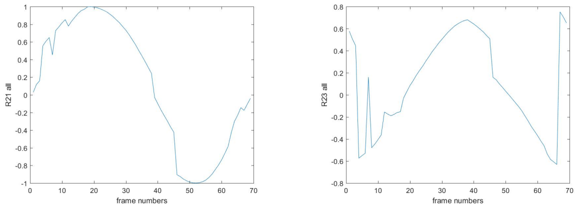


Fig. 3. Specific implementation process.

According to the Fig.6, the multi-view 8-frame image is filtered from the set of correct frames.



(a) The value of R_{21} per frame. (b) The value of R_{23} per frame.

Fig. 6.

B. Point cloud fusion reconstruction

1) Image Preprocessing - extracting the effective point Clouds: this article is aimed at the reconstruction of scene objects, so it is necessary to separate the background and get the effective point clouds. Processing directly based on the depth information is often unreliable. Therefore, the Yolov2 [13] algorithm is used to train and predict the color images to obtain the position information (bounding box) of the object in the images.

Algorithm 1 Truncated Signed Distance Field

```
for each $\{\mathbf{o_t}, \mathbf{x_t}\} \in Z_t$ do
    $z \leftarrow \|(\mathbf{o_t} - \mathbf{x_t})\|$
    $\mathbf{r} \leftarrow \frac{\mathbf{o_t} - \mathbf{x_t}}{z}$
    $\mathbf{v_c} \leftarrow (\mathbf{x_t} - u\mathbf{r})$
    $\tau \leftarrow \mathbf{T}(z)$
    // Update sdf $(\phi_\tau(\mathbf{v_c}))$ and new weight.
    for each $u \in [-\tau, +\tau]$ do
        $\phi_\tau(\mathbf{v_c}) \leftarrow (\frac{W(\mathbf{v_c})\phi(\mathbf{v_c}) + \alpha_\tau(u)u}{W(\mathbf{v_c}) + \alpha_\tau(u)})$
        $W(\mathbf{v_c}) \leftarrow W(\mathbf{v_c}) + \alpha_\tau(u)$
    end for
end for
```

2) Fusion and reconstruction: TSDF (Truncated Signed Distance Function) is a commonly used point cloud fusion method (Alg.1). Through the selecting frame, we can get the fusion points cloud with different angles. Due to the error of the camera acquisition device, the accuracy of the measured depth value is greatly reduced with the increase of the measurement distance. In view of this situation, this paper improves the quality of fusion by adjusting the weight and the fused image of each frame has its corresponding weight. Only the effective point cloud is used to establish and update the Chunk by Using Voxel Hash's structure and DDA algorithm, which greatly reduces the existing waste problem, and doesn't need to initialize the space size that makes the reconstruction more convenient. The entire fusion process takes 2 seconds.

3) In order to generate the mesh model, the Voxel Chunk is interpolated by the moving cube method to generate a rough triangular patch model, which is convenient for further processing of the point cloud.

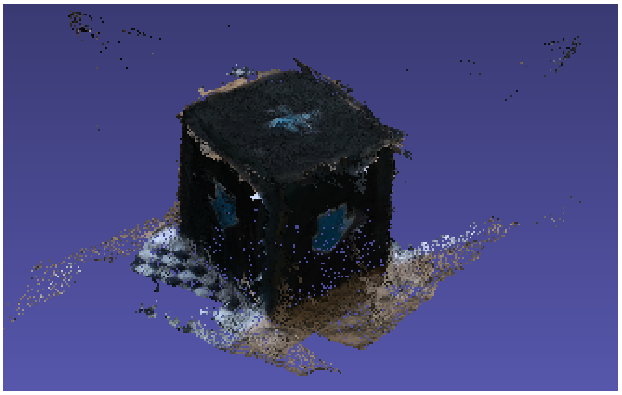


Fig. 7. Rough mesh by the TSDF method to fusion and MarchingCube method to generate.

C. Point cloud post-processing

According to the above reconstruction results (Fig.7), we can see that the model has many problems (such as the floating point, hole, surface roughness, etc).

1) Clear floating-points: according to the triangle connection mode, we can build the adjacency matrix and generate

a two-dimensional map. By using depth first search and dynamic planning, we find and save the largest connected area.

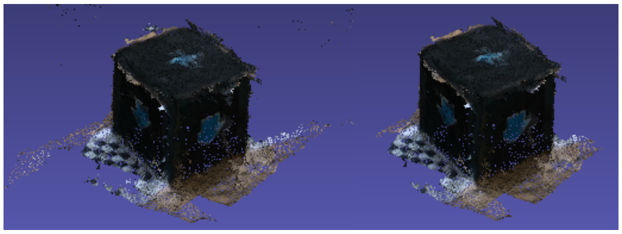


Fig. 8. Rough mesh VS clean mesh.

2) Smoothing: the commonly used Laplacian smoothing method [18] can achieve a good smoothing effect, but it will make the model size smaller. By referring to the relevant literature, this paper uses HC Laplacian smoothing method [19], and sets the two factors alpha and beta of the algorithm to 0.9 and 0.1 respectively to smooth the model, which not only achieves the effect, but also avoids the problem of model shrinkage (Alg.2).

Algorithm 2 HC Laplacian Smooth

```
$\mathbf{p} := \mathbf{o}$ // initialize original point
repeat
    $\mathbf{q} := \mathbf{p}$
    for all $i \in V_{var}$ do
        $n := |Adj(i)|$
        if $n \neq 0$ then
            $\mathbf{p}_i := \frac{1}{n}\sum_{j \in Adj(i)} \mathbf{q}_j$ // Laplacian smooth
        end if
        $\mathbf{b}_i := \mathbf{p}_i - (\alpha \mathbf{o}_i + (1-\alpha)\mathbf{q}_i)$
    end for
    for all $i \in V_{var}$ do
        $n := |Adj(i)|$
        if $n \neq 0$ then
            $\mathbf{p}_i := \mathbf{p}_i - (\beta \mathbf{b}_i + \frac{1-\beta}{n}\sum_{j \in Adj(i)} \mathbf{b}_j)$
        end if
    end for
until condition
```

3) Filling holes: after removing floating point and smoothing, we mainly solve the problem of filling hole. Through the literature review, we know that the poisson reconstruction is a good way to solve this problem (Fig.9). In this paper, the normal vector of each point is extracted and calculated from the depth image (2.5D) to ensure the consistency of the direction of the normal vector. The depth map has 2.5D information. So we can construct the mapping between z(x,y) (depth value) and pixel coordinates (x, y).

$$normal = (\frac{dz}{dx}, \frac{dz}{dy}, 1) \tag{4}$$

$$\frac{dz}{dx} = \frac{z(x-1,y) - z(x+1,y)}{2} \tag{5}$$

$$\frac{dz}{dy} = \frac{z(x,y-1) - z(x,y+1)}{2} \tag{6}$$

By constructing the water tightness equation (poisson's equation) [20], the new point cloud is generated by interpolation to fill the holes.

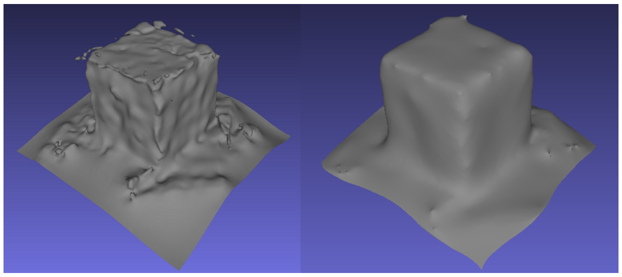


Fig. 9. possion VS possion+smooth.

Finally, we determine the plane of the maker by the reference 3D coordinates and trim the mesh. By extracting the texture from the original color image,we can obtain the final model (Fig.2b).

IV. CONCLUSION

Through the above-mentioned three-dimensional reconstruction algorithm framework, the feasibility of the scheme is verified by regular objects. We can also see the effect of post-cloud processing and improve the quality of model reconstruction (Fig.2b, Fig.7-9). In this paper, the model reconstruction of the object is realized on the consumer-grade hardware platform, which verifies the feasibility of the scheme and can balance both the reconstruction efficiency and the quality.

The basic theory of RGBD reconstruction has matured but further improvement is needed in the practical applications. The future work is to continue to optimize the algorithm and port it on the mobile end. The combination of deep learning and traditional 3D vision has always been the current research hotspot. So we will do further research on neural network and model compression so that it can be better applied in the field of 3D reconstruction and improve the quality of the model. The difficulty in reconstructing scene objects is how to achieve good background separation. Texture is an important factor affecting the visual effect of the model. How to render the model better and make the model more realistic is the focus of the future work.

REFERENCES

[1] R. Jamiruddin, A. O. Sari, J. Shabbir, and T. Anwer, “Rgb-depth slam review,” *arXiv: Computer Vision and Pattern Recognition*, 2018.

[2] R. Mur-Artal and J. D. Tards, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2016.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, “Orb-slam: A versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2017.

[4] A. E. Abdelhakim and A. A. Farag, “Csift: A sift descriptor with color invariant characteristics,” vol. 2, pp. 1978–1983, 2006.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: speeded up robust features,” *european conference on computer vision*, vol. 3951, pp. 404–417, 2006.

[6] Q. Tong, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[7] D. Kanoulas, N. G. Tsagarakis, and M. Vona, “rxkinfu: Moving volume kinectfusion for 3d perception and robotics,” pp. 1002–1009, 2018.

[8] M. Niesner, A. Dai, and M. Fisher, “Combining inertial navigation and icp for real-time 3d surface reconstruction,” pp. 13–16, 2014.

[9] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” *digital identity management*, pp. 145–152, 2001.

[10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison *et al.*, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” pp. 559–568, 2011.

[11] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” pp. 127–136, 2011.

[12] V. Lepetit, F. Morenonoguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[13] J. Redmon, S. K. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv: Computer Vision and Pattern Recognition*, 2015.

[14] M. Klingensmith, I. Dryanovski, S. S. Srinivasa, and J. Xiao, “Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields,” vol. 11, 2015.

[15] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” pp. 303–312, 1996.

[16] T. Whelan, S. Leutenegger, R. F. Salasmoreno, B. Glocker, and A. J. Davison, “Elasticfusion: dense slam without a pose graph,” vol. 11, 2015.

[17] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, “Facewarehouse: A 3d facial expression database for visual computing,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, 2014.

[18] D. A. Field, “Laplacian smoothing and delaunay triangulations,” *Communications in Applied Numerical Methods*, vol. 4, no. 6, pp. 709–712, 1988.

[19] J. Vollmer, R. Mencl, and H. Muller, “Improved laplacian smoothing of noisy surface meshes,” *Computer Graphics Forum*, vol. 18, no. 3, pp. 131–138, 1999.

[20] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Transactions on Graphics*, vol. 32, no. 3, p. 29, 2013.