Visual Odometry Based 3D-Reconstruction

Feng Jiang¹, Jianjun Gu¹, Shiqiang Zhu¹, Te Li*¹, Xinliang Zhong¹
¹Intelligent Robotic Research Center, Zhejiang Lab, Hangzhou 311121, China
*Corresponding author’s e-mail: lite@zhejianglab.com

Abstract. 3D reconstruction is a key technology for robots to explore unknown environments. The 3D model of the scene can be used to navigate the robot or used for scene segmentation. But when the scenes become larger or the features turns to rare, the frequently-used multi-scales matching and optimization algorithm of images and point cloud cannot work well. In this paper, we make the first attempt to split the 3D-reconstruction structure into 2 parts: pose estimation with the help of RGB images and reconstruction using poses from the aforementioned part and depth images from RGB-D cameras. All of these efforts are for accelerating the speed of pose estimation and promoting the precision of the reconstructed model, the results shows the proposed algorithm in the paper can do better than other reconstruction algorithms.

1. Introduction
With the continuous development of robotic technology, the ability of robots to accurately perceive the environment is becoming more and more important, among which the 3D-reconstruction, as a method to accurately perceive the environment, has gradually become a key method of all kinds of robots. 3D-reconstruction technology mainly uses RGB-D camera or other depth sensors to reconstruct a complete 3D model through the perception of the environment. The current mainstream 3D-reconstruction algorithms include BundleFusion[1] of Fusion series, ORB-SLAM[2] of VSLAM series and 3D-SIS[3] based on deep learning, etc.

However, 3D-reconstruction still faces many problems, the most important of which is the accuracy and speed of reconstructing. The main factors affecting the accuracy of modeling include the estimation of camera pose and the accuracy of depth information. The main factor that affects speed is the amount of data that needs to be processed.

In this paper, a new 3D-reconstruction framework combining visual odometry method is developed to solve the main difficulties of current methods. The 3D-reconstruction scheme is divided into two parts. The first part is used for pose estimation of the camera, which can be used for rapid calculation and iteration with general visual odometry(VO) algorithm. The second part uses the estimated camera pose data to perform dense 3D-reconstruction. The proposed method can shorten the processing time and improve the accuracy of reconstruction.

2. Overall
The overall block diagram of 3D-Reconstruction scheme based on the visual odometry is shown in Fig.1, which is divided into two main parts, the upper part for pose estimation: extract the features in the RGB image firstly, and then the features matching between different frames for interframe pose change, through the local and global optimization to obtain more accurate position. The lower part is the 3D-reconstruction part. Firstly, the depth map and poses are combined, and the depth data in the is
obtained by using VoxelHashing algorithm. Then, the 3D model in mesh format is obtained by using TSDF algorithm, and finally, the online display method is carried out by using OpenGL.

Fig.1 Overall of the visual odometry based 3D-reconstruction method

3. Details of the method

3.1. Camera Pose Estimation

Visual odometry (VO) is a common method to estimate the motion of cameras using only the input information from single or multiple cameras. VO obtains the RT transformation relationship between two adjacent frames, and multiplies the matrix to obtain the transformation relationship between the current frame and the original position, and also needs to optimize the result locally & globally.

Fig.2 shows an illustration of the visual odometry method. The relative poses of adjacent camera positions (or positions of a camera system) are computed from visual features and concatenated to get the absolute poses with respect to the initial coordinate frame at the origin frame.

In real environment, Kinect2 is used as an RGB-D camera to generate RGB images and depth images. Different from the traditional method, these two types of images are respectively fed into two branches that shows in Fig.1, wherein the RGB image is used for pose estimation and the depth image is used for 3D reconstruction.

Fig.2 An illustration of the visual odometry problem\[4\]
Firstly, ORB\(^5\) algorithm is used to detect features from RGB images. ORB features consist of two main parts: key points and descriptors. The key points is called Oriented Fast, which is an improved corner point of Fast, and its BRIEF is called Oriented Fast. The extraction of ORB features is divided into the following two steps:

1. FAST Extraction: Find the corners in the image. Compared with the original FAST algorithm, the ORB calculates the main direction of the features, adding rotation invariance to the BRIEF descriptor.

2. Brief Descriptor: Describes the pixel area around the key points found in the previous step. Since Brief is sensitive to image rotation, ORB has improved Brief by using the direction information calculated in the previous step to enhance the rotation invariance of Brief.

Secondly, after features are extracted from different images, features matching between two adjacent frames can be carried out to find the corresponding relationship of feature points between different images. The simplest matching algorithm is easy to understand: measure the distance of descriptors between each feature in picture I and all features in picture II, and then sort them to take the most proximate one as the matching point. The descriptor distance represents the degree of similarity between two feature points. For a binary descriptor such as BRIEF, it is common to use the Hamming distance: the offset of different bits in two binary strings.

Finally, loop closure detection method need to be used, the dictionary model is trained by k-means clustering (or hierarchical clustering, k-means), and the data structure of the dictionary model is tree structure. Find whether a word (that is, a feature) appears in the image, and use TF-IDF (frequency-inverse document frequency) as the value of a feature appearing in the vector. It can be judged by comparing the similarity of this vector between two images.

### 3.2. 3D-Reconstruction

![Fig.3 TSDF method of 3D-Reconstruction\(^6\)](image)

TSDF\(^6\) (Truncated Signed Distance Function), which is based on truncated signed distance function, is a common method to calculate hidden potential surface in 3D reconstruction. The famous Kinect Fusion uses TSDF to construct spatial voxels, by figuring out the value of each voxel, and then extracting surface using the Marching Cube algorithm.

\[
\text{TSDF}(p) = \frac{W(p)\text{TSDF}(p) + w(p)\text{tsdf}(p)}{W(p) + w(p)}
\]

TSDF is improved from SDF, and proposed the truncation distance. We will talk about the specific content below, which is very simple. In the case of parallel computing with large memory graphics card, the use of TSDF can achieve real-time reconstruction effect.
3.3. Online display

Another important step of 3D-reconstruction is online display. Using TSDF, we get a series of boundary values of points in the space. For online display, OpenGL method is selected, and triangular surfaces in the space are obtained through Marching Cube algorithm, and then the online display can be completed after input into OpenGL.

4. Test & Experiment

In order to test the actual modeling effect of the algorithm, TurtleBot was used to build a test platform. Data collection method is as follows: firstly, a suitable office room scene is selected as the target scene of reconstruction. After the installation of TurtleBot robot and Kinect2 camera, the robot chassis and Kinect2 camera are controlled and data recorded through Inter NUC. After the deployment, the room is reconstructed by controlling robot using keyboard.

Fig.5 establish the 3D-reconstruction result of the office room. On the left is the overview of the model, the model is reconstructed from half of the room, on the right of Fig.5 is a partial enlarged view of a table, from the figure you can see that the reconstruction results of the detail is effective, the complex office items can also be more accurately modeled, the overall average error of the model is less than 5cm.

After partially reconstruction testing, modeling testing and comparison of a larger scene are carried out. The whole room reconstruction result is shown in the Table 1. From Table 1 it shows that the difference between left figure and right figure. Left figure shows the result of the reconstruction result using algorithm in this paper, and the model from right figure is obtained by another SOTA algorithm. After comparison, it is found that the method in this paper can better restore and reconstruct the details of the scene, while the results obtained by the other algorithm is rough and the ability of detail
reduction is poor. Especially, our algorithm can well reconstruct the floor, while other methods cannot reconstruct the floor due to the matching mechanism, which is also one of the advantages of the algorithm in this paper.

| Our method | BundleFusion$^{[1]}$ |
|------------|-----------------------|
| ![Image](image1.png) | ![Image](image2.png) |
| ![Image](image3.png) | ![Image](image4.png) |
| ![Image](image5.png) | ![Image](image6.png) |

Finally, we also try to reconstruct a larger scene, the result is shown in Fig.6, the area of the whole exhibition hall is about 400m$^2$, and the reconstruction algorithm proposed in this paper has carried out a complete reconstruction of the scene and can restore the details of the scene better.
Fig. 6 3D-reconstruction result of an exhibition hall (about 400m²)

5. Conclusion
Based on the results and discussions presented above, the conclusions are obtained as below:

1. The proposed algorithm can separate pose estimation and 3D reconstruction, and can be used for multi-threaded optimization more effectively.
2. Visual information can be utilized efficiently while avoiding complex reconstruction point clouds affecting interframe matching.
3. Visual odometry allows for better reconstruction of planes with unclear textures, allowing for modeling of larger scenes.

References
[1] Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., & Theobalt, C. (2017). BundleFusion. ACM Transactions on Graphics, 762-770.
[2] Mur-Artal, R., Tardos, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33(5), 1255-1262.
[3] Ji Hou, Angela Dai, & Matthias Nießner. (2019). 3d-sis: 3d semantic instance segmentation of rgb-d scans. Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), 4421-4430.
[4] Nister, D., Naroditsky, O., & Bergen, J. (2004). Visual odometry. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR), 352-358.
[5] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer, 2564-2571.
[6] Werner, D., Al-Hamadi, A., & Werner, P. (2014). Truncated Signed Distance Function: Experiments on Voxel Size. Lecture Notes in Computer Science, 357-364.
[7] Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., & Newcombe, R. (2020). Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction. Computer Vision-ECCV 2020, 608–625.
[8] Nikoohemat, S., Diakité, A. A., Zlatanova, S., & Vosselman, G. (2020). Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. Automation in Construction, 113, 103109.