3D Object Modeling

Pin-Wei Chen[1]

I. INTRODUCTION & MOTIVATION

Nowadays, people are using 3D information [1] more often to do their work than 2D information. For example: self-driving vehicle, model showing, object recongnition, identity reconginition, etc. As a result, building a 3D model becomes a significant issue and technology, the result can offer to many field.

I am going to use only one depth camera (SR-300) to build a 3D point cloud model of an object. The way I build the 3D model is to move the camera around the object, and get lots of point cloud for different view, finally, I will merge all point clouds from different views to build the 3D model.

II. SYSTEM ARCHITECTURE & EQUIPMENTS

A. SYSTEM ARCHITECTURE

First, I use depth camera to recieve point cloud, and then I use TF to transform every point cloud to let their coordinate base on apriltags instead of camera link. Moreover, the way I get coordinate of apriltags is to use iSAM algorithm and get a more precise coordinate. And then, I use ICP [2] (Iterative Closest Point) to merge each point cloud properly. After that, we can build a 3D point cloud model of the object. The system diagram shows in Fig. 1.

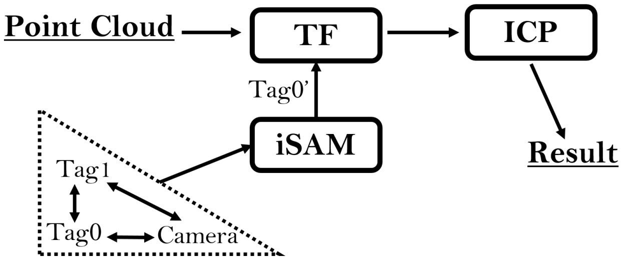


Fig. 1. 3D object modeling flowchart

B. EQUIPMENTS

For the hardware part, I use one depth camera (SR-300), which is connect to laptop, and I use several apriltags (36h11) to make the iSAM and TF work well. And then we need to put an object, which is going to be build a 3D model Fig. 2.

In software part, we use ROS kinetic, OpenCV 3.2.0 and ubuntu 16.04 LTS as our develop environment.

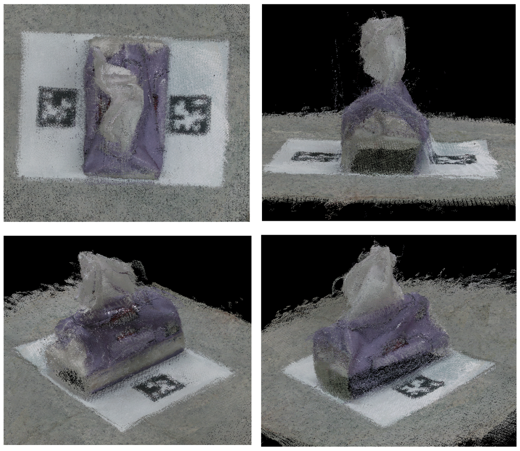


Fig. 2. 3D object modeling

III. APPROACH

- Using iSAM [3] to get a better performance of coordinate

I use several apriltags to get lots of coordinate informations, ant this can help to increase the amount of landmark for iSAM algorithm.

First, I fix every apriltags, and measure their relative position, and these relative position will become apriltags-apriltags pose3d parameters. Then I will use depth camera to capture all apriltags, getting the relative pose between camera link and apriltags, and these will become apriltags-camera pose3d parameters. At last, I add those relations, including apriltags-apriltags and apriltags-camera link, to iSAM pose3d factor, and call the function to optimize the coordinate, I will do this step once I get an point cloud from depth camera.

The aim of this step is to optimize the coordinate of apriltags, because the coordinate of apriltags only get from camera will have some errors, so I use iSAM to reduce the error.

- Transfer coordinate by TF

TF is a library for ROS, and the function is to transfer the point cloud coordinate from camera link to apriltag, it can help us unify the coordinate from different view of point cloud.

The input of the TF is a point cloud and two frame, one is origin frame and another is target frame, and we can get the oput point cloud that have already transfer it coordinate from origin frame to target frame.

The aims of this step is to let every point cloud base on the same frame, so that all point clouds can approximately

*This work was supported by the Robotics Master Program in National Chiao Tung University, Taiwan

[1] Pin-Wei Chen, National Chiao Tung University, Taiwan. ccpwearth@gmail.com

match together.

- Merge point clouds using ICP

The coordinate of apriltags generate from iSAM will still have some error, so that the point cloud will not match with each other prcisely, so I use ICP to help merge each point cloud together perfectly.

ICP is the abbreviation of ”Iterative Closet Point”, this library can use iteration to rotate and translate one point cloud to match another point cloud.

- Build a 3D object model into PCD file

After merging all the point cloud, I can get the 3D model of the object, and I save it as an PCD file, so that others can use this model conveniently, moreover, there are lots of library and editor such as PCD editor which can adjust and edit the point cloud easily.

IV. RESULT

- Untreated Point Cloud

The original untreated point cloud is generated by adding the whole point cloud directly, however, every point cloud base on different frame, so the result will be pretty messy Fig. 3.

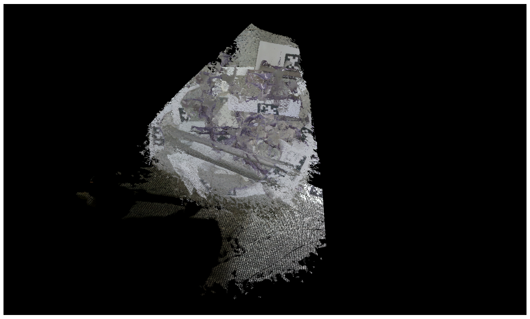


Fig. 3. Untreated Point Cloud

- Using TF

The result of using TF looks much more better then untreated one, because every point cloud have already base on same frame, apriltags ID: 1, so that they can match approximately, however, the coordinate of the apriltags still have some error, so they still can’t match perfectly Fig. 4.

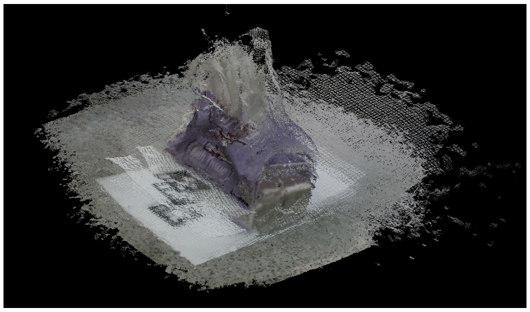


Fig. 4. Using TF

- Using ICP

When only use ICP to merge the point cloud, it takes a very long time to finish the merge of two point clouds. It is because the two point clouds are base on two different coordinates, the ICP algorithm need lots of iterations and caculations to merge the two point clouds. As a result, during the limited time, the final result of only using ICP can merge will but lack of points, due to the long caculation time. Fig. 5.

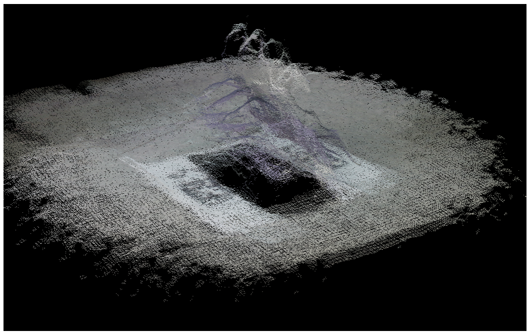


Fig. 5. Using ICP

- Using TF + ICP + iSAM

Combining the result of above, I use TF and ICP together to improve the performance, and I also use iSAM algorithm to optimize the apriltags coordinate. The result of Using TF + ICP + iSAM is pretty satisfying.

Now, it can merge more precisely then only using TF. Moreover, it has more points then only using ICP, because the whole point clouds have already using TF to base on the same coordinate, the ICP caculation time and iteration will reduce significantly. As a result, I can get a result that has perfect merging and enough points to see the details of the object Fig. 6.

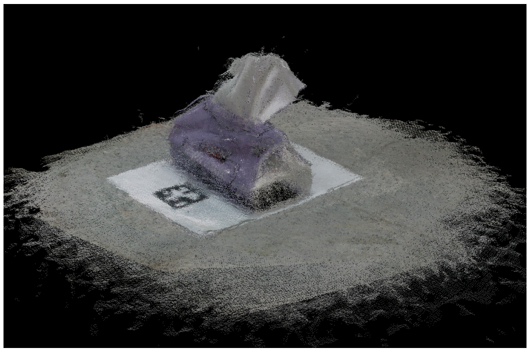


Fig. 6. Using ICP

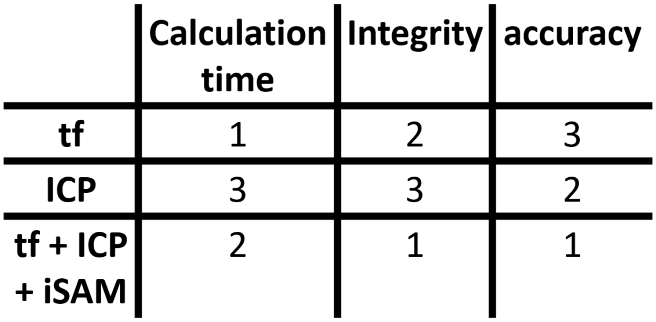
V. CONCLUSION

We can see the comparasion table in Fig. 7, the calculation time TF works the best, cause it doesn’t really need any caculation, the only caculation is the transform of the coordinate, and the worst case is using only ICP, because it needs lots of iterations.

For the Integrity, the worst case is using only ICP, because it need lots of caculation time and iterations, so it will lack of points, making lots of details of object been disappear. The best case is using TF + ICP + iSAM.

For the accuracy, the best case is using TF + ICP + iSAM, because it transfer the coordinate and also use ICP to merge the point cloud, and the worst case is using only TF, because the error of apriltags will always exist.

In conclusion, the best performance is to use TF + ICP + iSAM, it can generate a complete point cloud model which is been merge pretty well.



	Calculation time	Integrity	accuracy
tf	1	2	3
ICP	3	3	2
tf + ICP + iSAM	2	1	1

Fig. 7. Comparasion

REFERENCES

[1] G. Vosselman, S. Dijkman, *et al.*, "3d building model reconstruction from point clouds and ground plans," *International archives of photogrammetry remote sensing and spatial information sciences*, vol. 34, no. 3/W4, pp. 37–44, 2001.

[2] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[3] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.