DEVELOPMENT OF AN EFFICIENT RGB-D ANNOTATION TOOL FOR VIDEO SEQUENCE

by

BAOZHANG REN

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Computer Science

Written under the direction of

Kostas Bekris

and approved by

_____

_____

_____

New Brunswick, New Jersey

January, 2020

ABSTRACT OF THE THESIS

Development of an Efficient RGB-D Annotation Tool for Video Sequence

By Baozhang Ren

Thesis Director:

Kostas Bekris

With the prevalence of neural networks and deep learning models, more data is required to expand the domain as well as to improve the accuracy of those models. There are numerous annotation tools and software of RGB data aiming to make the labeling process less gradual and more efficient while maintaining the same accuracy as traditional methods. However, fewer such efforts have been made in the RGB-D domain. This paper provides a novel RGB-D annotation tool that is designed to efficiently generate object poses in images or video sequences. The tool is equipped with functions, such as removing background points, interactive marker, to aid annotation, as well as ICP to lower the amount of frames that need to be labeled in a video sequence.

Acknowledgements

Many thanks to Bowen Wen for the help of the collection of data and ICP implementation.

Table of Contents

List of Figures

Chapter 1

Introduction

Neural networks, which is one of the most popular algorithms for numerous task, such as object detection and action recognition, are hungry for data. As algorithms and machine learning models grow deeper as well as the problems that needs to be solved become more complex, the amount of data required for the models to perform ideally needs an explosive increase.

For object detection and recognition task, the renowned datasets, ImageNet and Microsoft Common Objects in Context (COCO), have more than 14 million and 2.5 million labeled instances respectively[1, 2], which requires huge amount of work and efforts. According to [3], the initial version of the ImageNet dataset has only 3.2 million labeled images and it still took two and a half year to finish with the help of Amazon Mechanical Turk. The task gets harder when it comes to video labeling. For human pose tracking, one of the most popular dataset, Posetrack dataset, has 550 labeled video sequences, 66,374 frames, 153,615 labeled poses and about 2.7 million labeled joints [4]. Different from image object detection and recognition, it takes more labor to label ground truth data of video sequences since the label difference between continuous frames needs to be small and smooth.

Manually labeling all the data is time consuming. According to [5], it takes 1140 seconds to label one image from COCO+stuff data for crowd sourced annotators. And for human experts annotating on tools like LabelMe, it takes about 507 seconds. It means that to label the whole dataset, 52,000 hours are needed for crowd sourced annotators and 23,000 hours are needed for human experts, which is not affordable for some small companies or individuals. Therefore, various researchers have proposed annotation tools, like Fluid Annotation by Google, that equipped with deep learning

models to help annotators and shorten labeling hours.

For labeling task of RGB-D images in video sequence, it is required to consider both location and the rotation of a given object. Therefore it rises several problems that make it a more complex task than tasks mentioned before. First, it is hard to locate the exact location in three dimension, especially when the RGB-D still has only a two dimensional view of the object. Second, occlusion makes it hard for algorithms, like iterative closest point(ICP) to perform semi-automatic labeling when there is occlusion to the object. Third, if it is possible that the tool labels data accurately without reconstructing the scene using sensor fusion or using motion capture. A RGB-D labeling tool is developed to address these three problems. For the first problem, the tool is equipped with ICP to auto correct pose labeled by human in order to minimize the error. The reason for the second problem to happen is that when the object is occluded, the majority of the points around the previous pose do not belong to the object. Thus, the tool is equipped with a point remover to remove unrelated points. It is tested that ICP can perform accurately after removing background points in highly occluded cases. To address the third problem, the tool can be used when there is only one RGB-D image per frame. For fast labeling, for each frame, a suggested pose is presented on the screen by ICP. Since the results of ICP for nearby frames are usually correct, human annotators can skip frames when there are only small movements. Also, the tool is build on Rviz and Qt and can easily customize to other types of labeling problems.

Chapter 2

Literature Review

Realizing the difficulties of creating such data, annotation tools that are easily accessible and efficient have been made. In order to better understand what have been done and the problems we are still facing, the following two sections will include various labeling tools of RGB as well as RGB-D images and videos.

2.1 Image and Video Labeling Tools

A commonly used annotation tool LabelMe[6] was created in 2005 in order to facilitate computer vision researcher with a tool to share and label images. It has accumulated over 400,000 labeled objects as of 2009. It support both polygon and bounding box for spacial content labeling. However, it is slow since it does not have algorithms to provide human annotators with label suggestions.

VGG Image Annotator (VIA)[7] is a standalone annotation package that can run solely on a web browser. It allows human annotators to provide semantic meanings to selected region of images, videos and audios and to retrieve the data in plain text formats like JSON. The tool supports semi-automatic labeling in two stages in a group of images, automatic annotation provided by background computer vision algorithms like Faster R-CNN, and manual filtering by human annotators to filter, select and update automatic annotations. It is useful in identifying unique individuals in a image or video sequence.

Fluid Annotation[8], an human-machine annotation interface developed by Google, can be used to label full a image in a single pass with the help of strong machine-learning aids. The interface requires predefined set of classes that need annotation. Then a active set of non-overlapping masks, which act as the initial annotations, is provided by

a neural network. At last, human annotators can carry out a series of actions, such as changing mask label, adding or removing segments to the set and changing the depth order of the set, to generate accurate ground-truth data. This interface is tested to be 3 times more efficient than the LabelMe interface on the COCO+Stuff dataset.

Other than asking computer algorithms to provide annotation automatically, [9] suggested a way to create segmentation of an object by providing four extreme points on the object. In [10], it showed that using extreme clicking points to generate bounding boxes is 5 times faster than traditional methods while the quality of the boxes as well as the accuracy of a model trained on thoses boxes are as good as using ground-truth data gathered in original ways. Deep Extreme Cut(DEXTR) leverages output points and inferred bounding box using [10], as input, ResNet-101 as backbone and Deeplab-v2 model pre-trained on ImageNet to get the semantic segmentation of a selected object. The labeling cost for a single mask can be reduced form 79 seconds to 7.2 seconds. The masks produced using this methods are proven to have the similar quality as those generated by traditional methods.

2.2 RGB-D Labeling Tools

[11] showed an object oriented 3D Slam paradigm, SLAM++, which can estimate and track pose of a given object type relative to a fixed world frame and store historical camera pose in each time step. Given 3D model meshes of objects, SLAM++ utilized research of Drost et al.[12] to recognize 6 DoF poses of 3D objects by Point Pair Features(PPFs).

Nguyen el al.[13] proposed a 3D-2D interactive annotation tool aimed to enable faster labeling in a dense 3D scene. The tool first constructs a 3D scene mesh from RGB-D frames and match correspondent 2D pixels and 3D vertices for fast 2D and 3D segmentation switches. In the second step, the scene is automatically segmented using graph-based segmentation, dividing it into supervertices, and MRF-based segmentation, which merges supervertices using surface normals and colors. Then, users can refine those segments by merging nearby segments, extracting supervertices from segments

and spliting severe under-segmentations.

Marion el al.[14] suggested a novel pipeline, LabelFusion, to quickly generate large amount of high quality RGB-D data with pixelwise labels and object poses. The pipeline consists of five stages. First, multiple RGB-D frames are generated by hand-carried or automated arm-mounted cameras. Second, reconstruct the 3D scene from the collected RGB-D data, using ElasticFusion[15], which also provides camera pose tracking relative to the local reconstruction frame. Next, the third stage is to obtain meshes of the objects using scanners. The final stage requires human annotators to provide rough initial poses for each object by 3 clicks on the reconstructed scene points and on the object mesh. Then the program specifies an initial alignment of the mesh to the scene and cropped a pointcloud from the scene based on the alignment. Finally, the program perform ICP to refine the initial pose. Labels of later frames will then be automatically generated based on the camera pose from stage two.

Other than ICP, researchers proposed other algorithms to perform object pose estimation. In [16, 17], they used neural networks trained on RGB images to locate and crop out point clouds of selected objects. Then utilize ICP to estimate and align poses between cropped point clouds and object meshes.

Compared to RGB image labeling tools, RGB-D labeling tools have less varieties. Most of the RGB-D labeling tools are used in a controlled environment and can not adapt to changing conditions. [11, 13, 14] are only used when the camera moves and the objects remain stationary and many of the tasks nowadays requires moving objects and stationary cameras. In [14], they need reconstructed 3D scenes which will incur errors in reconstructions. Since the output poses rely on those scenes and the program provides no methods to alter the outputs of the program, they will have larger errors compare to human labeled poses. Pipelines provided in [16, 17] do not have reconstruction errors and are suitable for moving objects. However, training a neural network that can identify those objects also requires data, which still demands large amount of annotations, and it is especially difficult for objects with no texture and no distinguishable colors and shapes. The tool shown in this paper can work on both stationary and moving objects and it only requires RGB-D data and object meshes. The tool can refine

object poses on every frame as well as providing semi-automatic labeling suggestions from ICP to minimize human efforts.

Chapter 3

Label Generation Tool and Pipeline

The main contribution of this paper is a highly customizable tool GUI and an efficient labeling pipeline for RGB-D data. Section 3.1 will briefly introduce Iterative Closest Point(ICP) algorithm, which is one of the main component of the tool. Section 3.2 will show components that the tool was built on. In section 3.3, the pipeline and main functions will be discussed.

3.1 Iterative Closest Point

Iterative Closest Point[18, 19] is widely used to find transformation of two relatively close point clouds by minimizing the root squared error between the source point cloud and the reference point cloud. The algorithm requires source and reference point clouds and stopping criteria as input. Although it is optional to provide an initial transformation from the source to the reference for input, a good initial transformation will speed up the process and prevent convergence fail of the program. Given the inputs, the essential steps are as follows[20]:

1. For each point in the source point cloud, find the closest point in the reference point cloud.
2. Estimate the transformation that best match the source points and their matches from previous step using root squared error.
3. Transform source points using the obtained transformation matrix.
4. Iterate the process from step 1 to 3.

The ICP algorithm is useful in two cases in the pipeline, to refine human annotated

poses for more accurate annotation and to predict next pose based on current pose of a certain object in order to reduce human labor.

3.2 Label Generation Tool

The tool is built on ROS Visualization(Rviz) and customized using Qt[21]. Rviz is a 3D visualizer that are commonly used to display point clouds, sensor data and state information from ROS. Qt is a widget toolkit for creating and customizing graphical user interfaces or cross-platform applications. All the functions shown below is connected to Qt widgets.

3.3 Label Generation Pipeline

Figure 3.1 exhibits the graphical interface of the tool and figure 3.2 shows the pipeline of the tool .

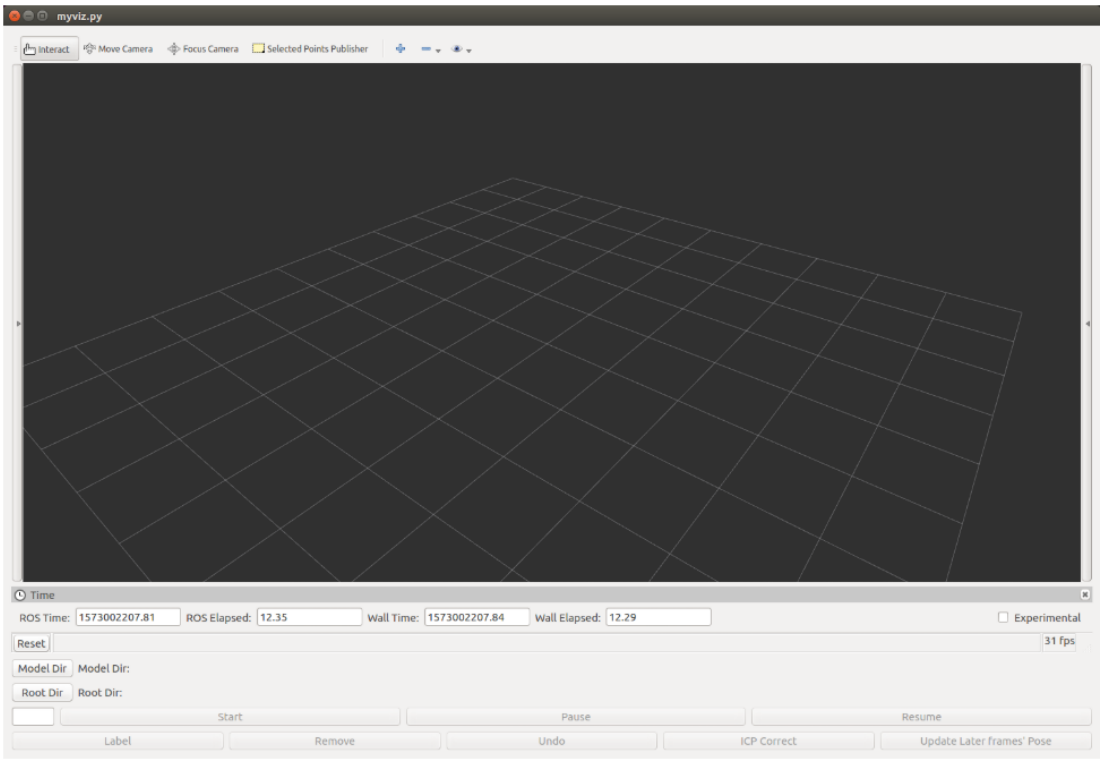


Figure 3.1: Graphical Interface of the Tool

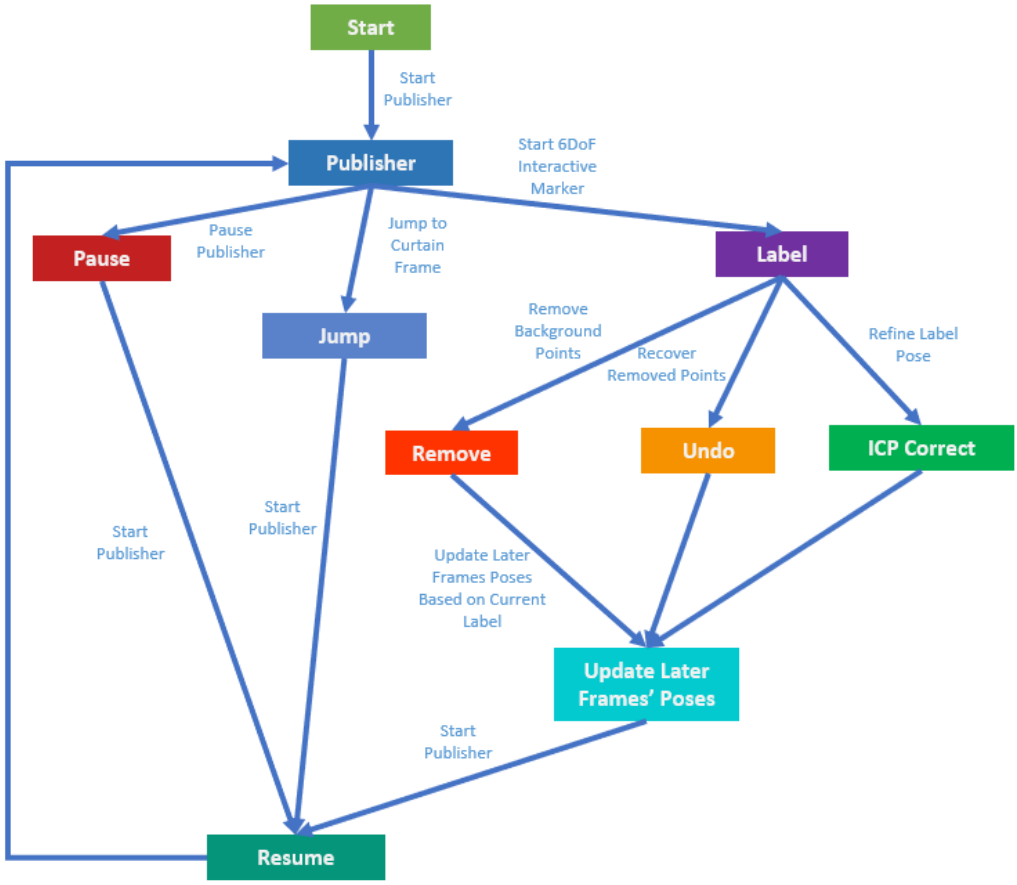


Figure 3.2: Pipline of the Tool

The tool has nine functions and they will be demonstrated in the sections below.

3.3.1 Start

Before pressing the start button, two directories should be provided to the tool. The first one is Model Dir, which is the directory path to the mesh of the object. The second one is Root Dir, which is the root directory of the whole data set. In the root directory, there should be two folders, a ply folder that includes the point clouds of each scene and a icp_annotated_poses folder that contains transformation matrices of the model in each scene. If the icp_annotated_poses folder is empty, the initial transformation is a 4×4 identity matrix.

After selecting the directories, press the start button. The point clouds of the scene and the transformed model point clouds will be published to the topics /scene and

/model respectively. Figure 3.3 shows an example of the point cloud visualization. The white dots indicate current model's pose in current frame. Users can pause the publisher, jump to certain frame or go straight to labeling.

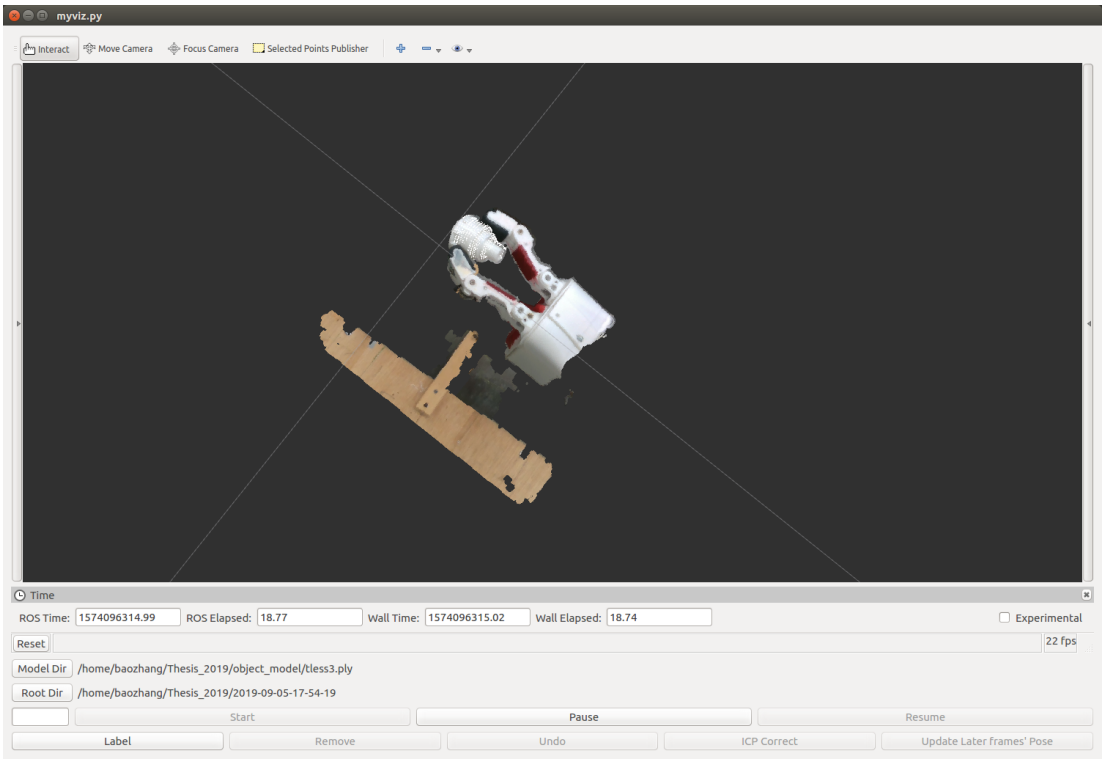


Figure 3.3: Start

3.3.2 Pause

By pressing the Pause Button, the program will stop publishing to the two topics. From this point, the program can resume publishing by pressing the Resume button, jump to certain frame, or start labeling by pressing the Label button as demonstrated in figure 3.4.

3.3.3 Jump

Users can enter frame number in the slot next to the Start button like in figure 3.5. When the ICP program is updating Poses, users can only jump to frames that have been updated.

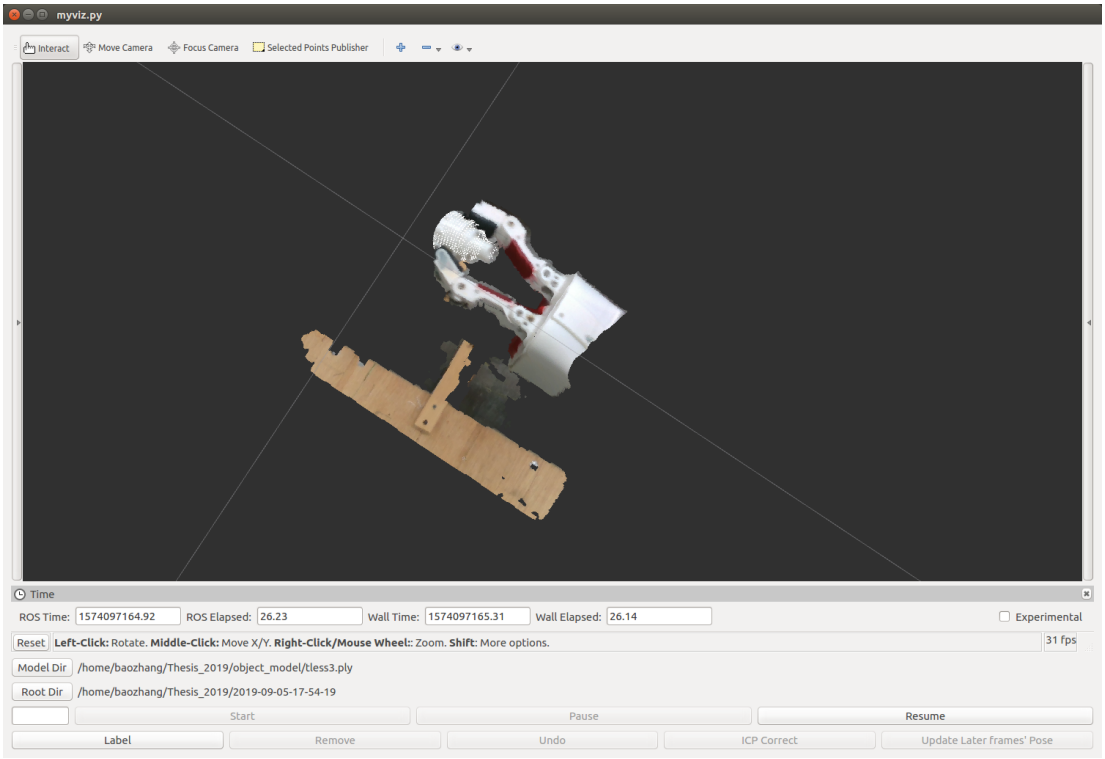


Figure 3.4: Pause

3.3.4 Label

To label the current frame, press the label button and an interactive marker will pop as in figure 3.6. Users can move the model using the arrows and rotate the model using the rings. The new pose will be recorded but not saved until Update Later Frames' Poses button is pressed.

3.3.5 ICP Correct

In order to reduce human annotation errors, an ICP Correct button can be used to run ICP on the label model point cloud and the scene point cloud. If users removed points from the scene, ICP correct will run ICP on the label model point cloud and the remain scene point cloud. An example is shown in figure 3.7.

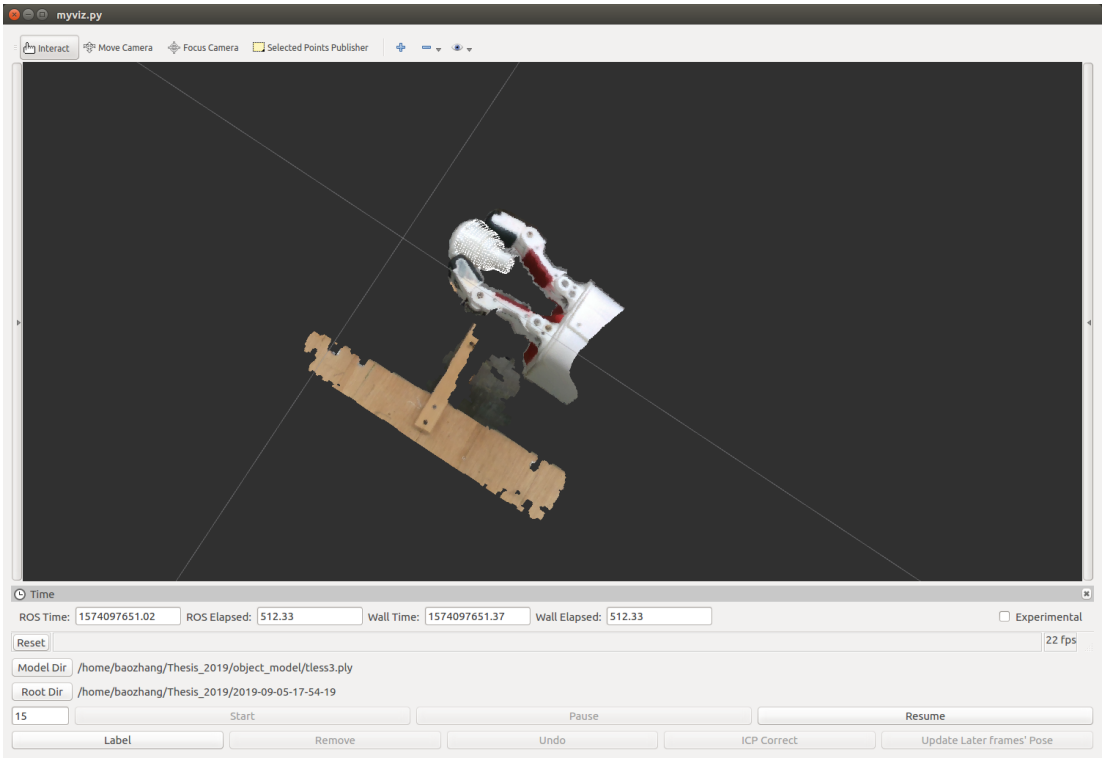


Figure 3.5: Jump

3.3.6 Remove

while labeling frames with the highly occluded object, ICP Correct doesn't work well or even doesn't converge because of the noise incurred by the background points in the scene. Utilizing Selected Points Publisher provided by turbo-ros-pkg[22], users can select points in the region and publish them to the topic /rviz_selected_points. The program will subscribe to the topic and remove selected points. An example is shown in figure 3.8.

3.3.7 Undo

Before Removing selected points, current point cloud is stored. Users can recover removed points by pressing the Undo button and the publisher will publish previously stored point clouds.

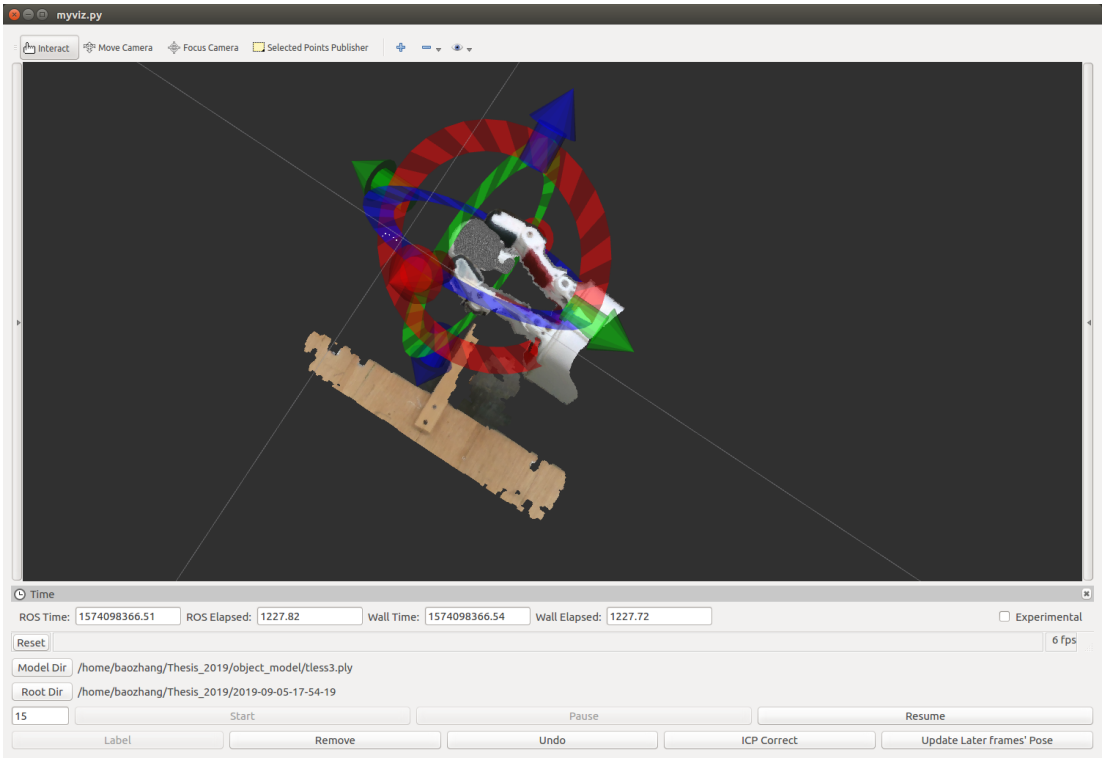
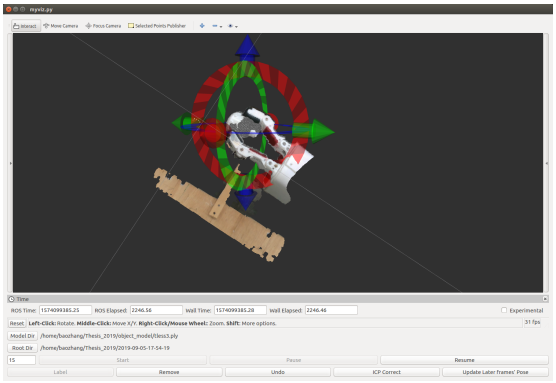


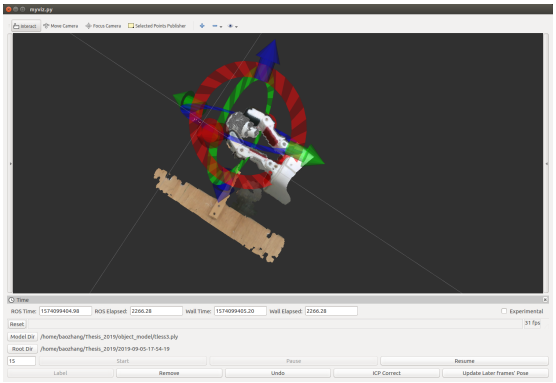
Figure 3.6: Interactive Marker

3.3.8 Update Later Frames' Poses

This function will update all later frames' poses based on the label of the current pose using ICP. After pressing the Resume button, the publisher will start again. However, the publisher will only publish pose after it is updated by ICP. The program will be stopped when the Label button is pressed and will be resumed when the button is pressed again.

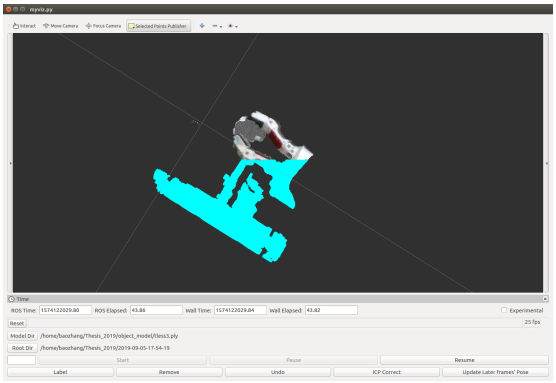


(a) Before ICP Correct

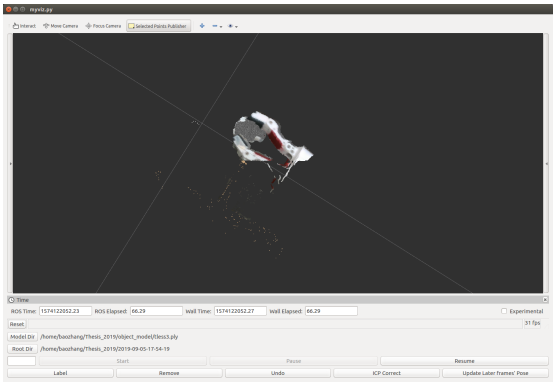


(b) After ICP Correct

Figure 3.7: ICP Correct



(a) Select Points



(b) Remove Points

Figure 3.8: Remove

Chapter 4

Discussion and Conclusion

This paper introduces a novel tool to label RGB-D data in a frame by frame basis. The pipeline provided is easy to follow and is very efficient when labeling video sequences.

Since the tool is built on Rviz and Qt, it is easy to add or change background functions. For example, the ICP function can be substituted to other registration or tracking algorithms and users can use the tool for more accurate labeling or simply a visualization tool for testing. It can also adapt to scenes with 3D reconstruction, or enable multiple object labeling in a single pass.

We hope that this tool can lower the bar for RGB-D data labeling as well as simplifying its visualization process by demonstrating the simple tool interface and pipeline.

Bibliography

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, December 2015.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[3] Dave Gershgorn. The data that transformed ai research-and possibly the world, Jul 2017.

[4] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking, 2017.

[5] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. Fluid annotation. *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*, 2018.

[6] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008.

[7] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. *Proceedings of the 27th ACM International Conference on Multimedia - MM '19*, 2019.

[8] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. Fluid annotation. *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*, 2018.

[9] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation, 2017.

[10] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation, 2017.

[11] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *CVPR*, pages 1352–1359. IEEE Computer Society, 2013.

[12] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, pages 998–1005. IEEE Computer Society, 2010.

[13] Duc Thanh Nguyen, Binh-Son Hua, Lap-Fai Yu, and Sai-Kit Yeung. A robust 3d-2d interactive tool for scene segmentation and annotation, 2016.

[14] Pat Marion, Peter R. Florence, Lucas Manuelli, and Russ Tedrake. Labelfusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes, 2017.

[15] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[16] Jay M. Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David M.S. Johnson, and et al. Segicp: Integrated deep semantic segmentation and pose estimation. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017.

[17] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker Jr., Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge, 2016.

[18] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[19] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, April 1992.

[20] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3DIM*, pages 145–152. IEEE Computer Society, 2001.

[21] The Qt Company. Qt-about us. `https://web.archive.org/web/20170222172844/https:/www.qt.io/about-us/`.

[22] tu-rbo/turbo-ros-pkg. `https://github.com/tu-rbo/turbo-ros-pkg`, Mar 2015.