Real-time point cloud construction and information system

Firebot Telepresence

A VR application that constructs a point cloud based on the information of the Firebot

Project Information

Members:

- Willem Lefferts - 449870@student.saxion.nl
- Andrei ‘Jesse’ Martashov - 448151@student.saxion.nl
- Stan Smeitink - 418461@student.saxion.nl
- David Heetebrij - 437445@student.saxion.nl

Client:

- Danny Plass - Oude Bos - d.plass@saxion.nl

Duration: 10/02/2020 - 10/07/2020
Research Group Ambient Intelligence

Methodology

Our main research methodology was desk research and prototyping.

Desk research is another name for secondary research. Broadly speaking, secondary research is where you review what other people have done.

Prototyping is an experimental process where our team implemented ideas we came up with during our desk research.

These experiments gradually evolved into our final application.

Future Work

While we use ROSbridge with a WebSocket connection, ROSbridge itself is not tied to any particular transport layer. It might be possible to add a ROSbridge TCP package or node, for example, that communicates with ROSbridge using TCP sockets. This way you do not need to have both the machines on the same local network and they can have some distances in between them. We also want to encourage more in depth research into handling an actual real time stream instead of pre recorded data. We do not expect any issues with connecting and getting data across. However, at that point it would be necessary that the data builds up the point cloud from the data it gets. Right now it is more similar to a video and the point cloud does not increase in size. This should be different in the eventual application.

The current data structure we use for sending point clouds may be too large for the Oculus Quest to handle. We have not found a suitable alternative because we did not have the time. However, finding a better way of transferring data should improve the performance of the application greatly.

Research Questions

How can we implement a point cloud in VR with Unity?

- Unity ROS# plugin to stream point-cloud data from ROS to Unity
- Unity XR plugin to use VR view and controls
- What optimizations are going to be needed?

How to visualize data from ROS to Unity?

- How can we get point cloud data from the raw data?
- How can we get the data from ROS to Unity?
- How can we show the point cloud in Unity?

Conclusion

Visualize data from ROS to Unity:

To set up a connection between ROS and unity we use ROS# in combination with ROSbridge and a point cloud subscriber. This works very well and is easy to set up, especially if you know how to use ROS properly. One thing we are not sure about is how flexible the connection is. We have only tested it with the following configuration:

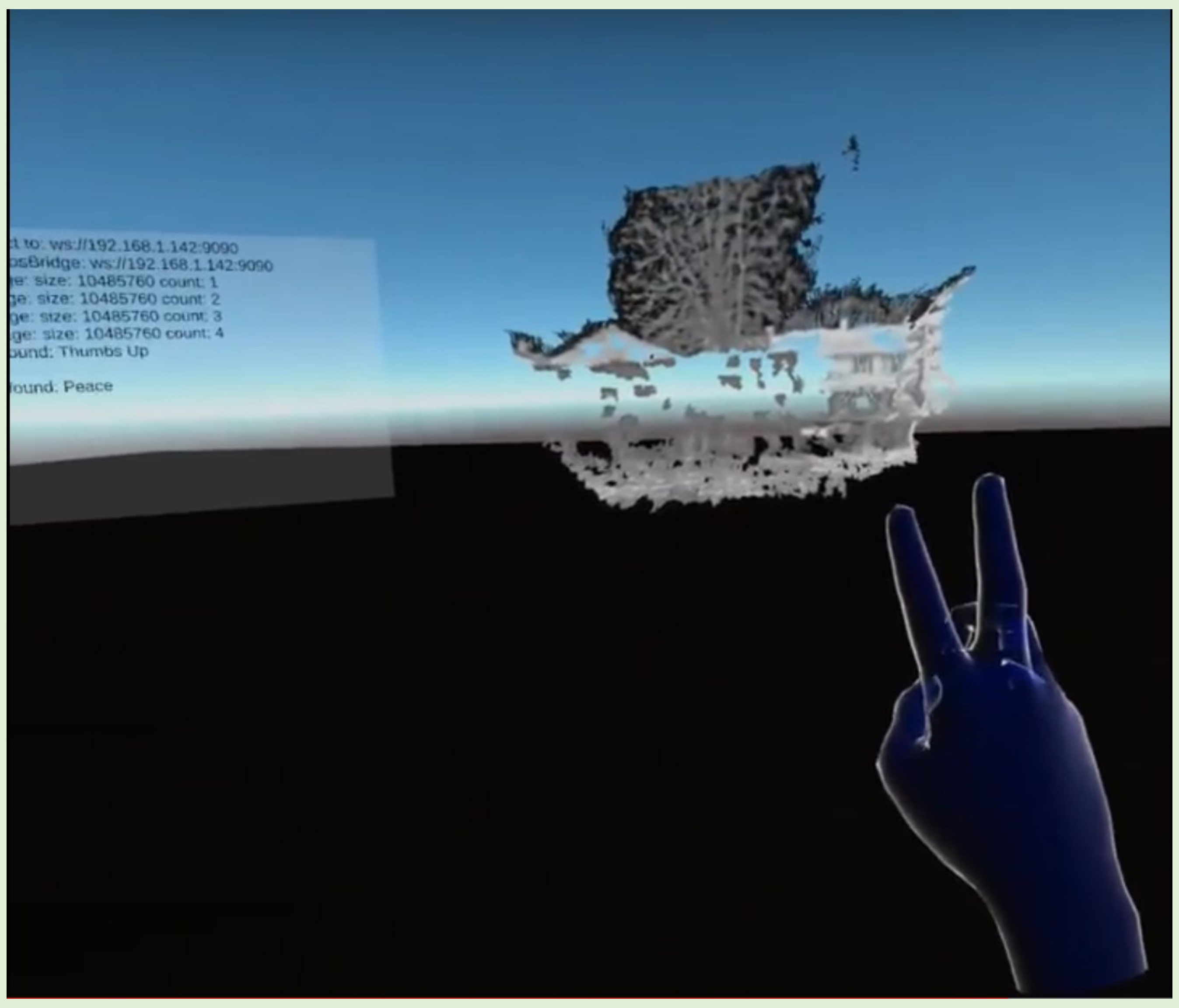
- A PC with Unity
- A Ubuntu virtual machine that runs on the same PC
- The data is pre-downloaded on the Ubuntu machine

When the ROS side eventually gets from the robot, the robot will need to maintain connection with the VR headset or the machine on which the VR application is run.

Implement a point cloud in VR with Unity:

Point Cloud data is hard on a system’s performance. We have tried to reduce this impact, and have succeeded in coming closer to a finalized product. However, due to the nature of the target platform which has limited capabilities, and also requiring it to be run in virtual reality, these methods, while allowing for a significant increase in performance, may not be enough.

Still, we believe that at the very least we have provided future project teams a sizable foundation upon which to expand, and improve, so that an acceptable product can be reached.



```
t to: ws://192.168.1.142:9090
osBridge: ws://192.168.1.142:9090
ge: size: 10485760 count: 1
ge: size: 10485760 count: 2
ge: size: 10485760 count: 3
ge: size: 10485760 count: 4
ound: Thumbs Up

ound: Peace
```