Remote Robot Car Control System with RGBD Camera for 3D Reconstruction

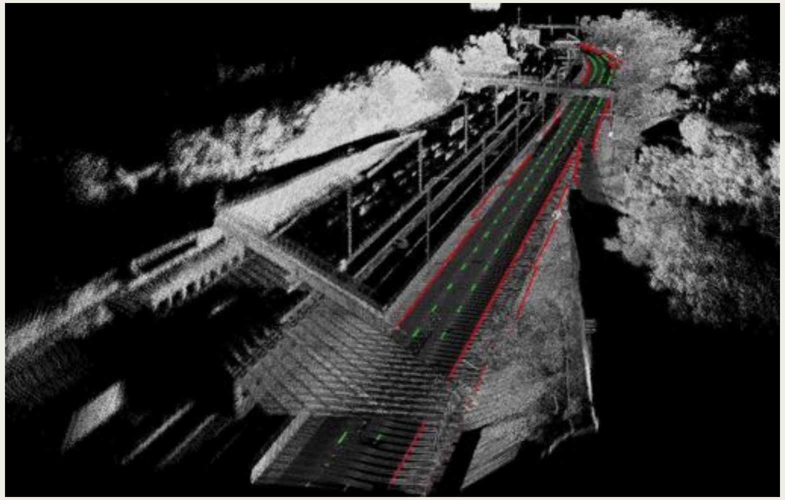
Team#21 Members
Yuhao Ge, Junyan Li, Hao Chen, Han Yang

Teaching assistant: Yiqun Niu
Sponsor: Prof. Pavel Loskot

ECE445 Senior Design
SP23 ZJUI TEAM#21

Problem Statement: Model is Everywhere

Road Mapping



Augmented/Virtual Reality

Game Modeling

Historical Site Digitization

Problem Statement: Challenges

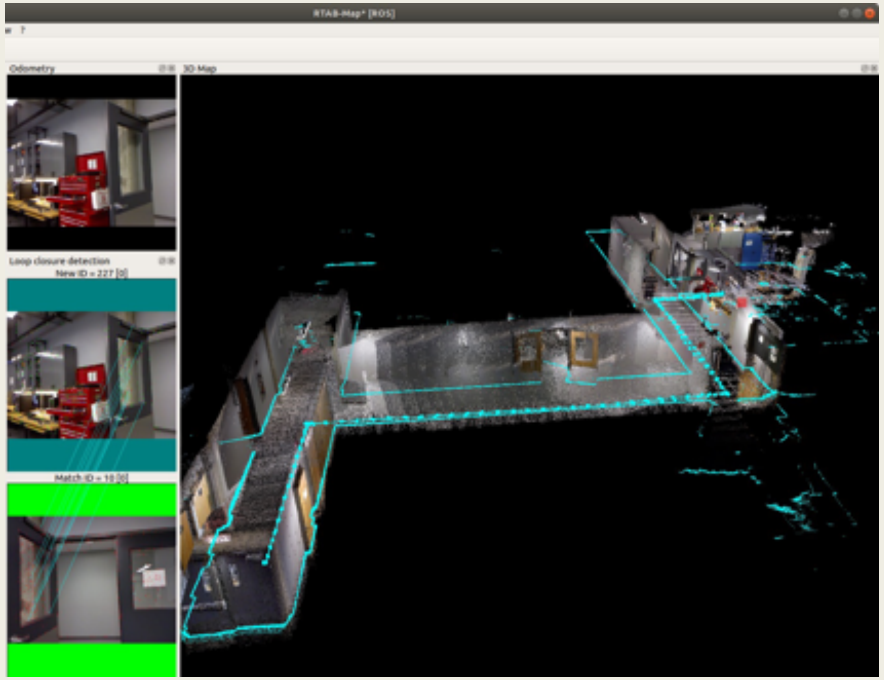
- Flexibility and Safety
  - *Some sites are dangerous*
  - *Some places are difficult to access by human*
  - *Humans are lazy*

- Feasibility
  - *Computational payload is high for 3D algorithms*
  - *Edge devices should be small and have low power consumption*

Our Design Focus on

Remote Vehicle Control

Remote Realtime 3D Reconstruction



Combine these Two Functions!

Our Solution

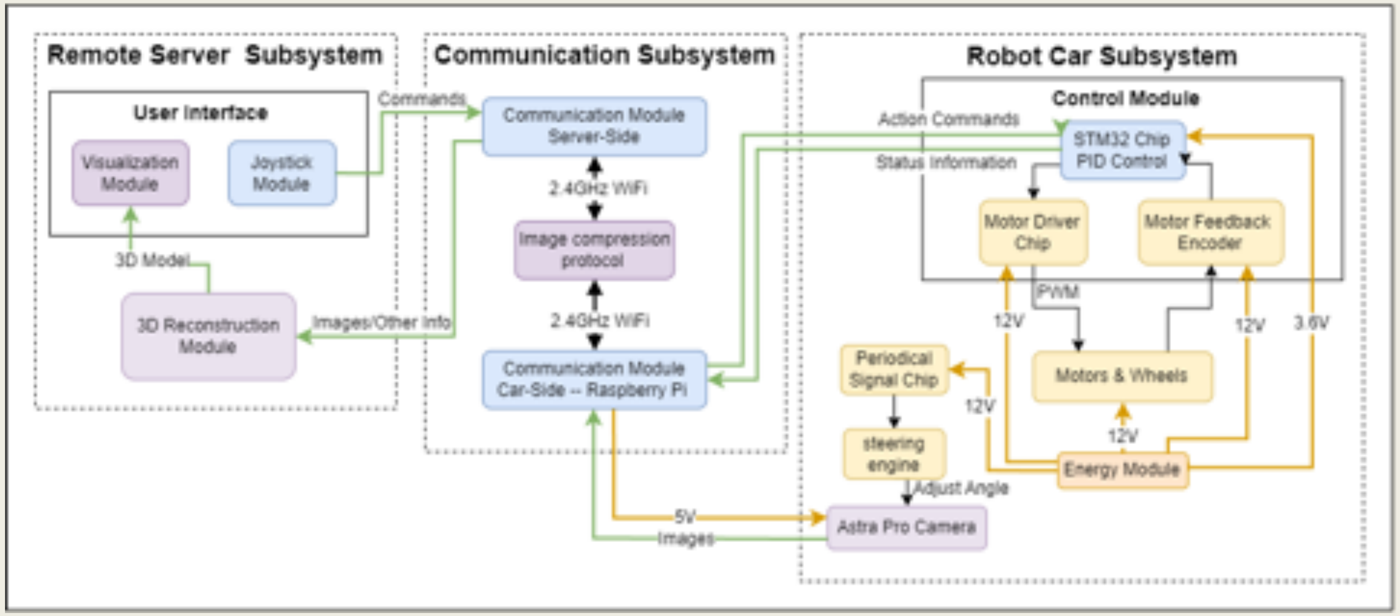


Figure 1: Block Diagram

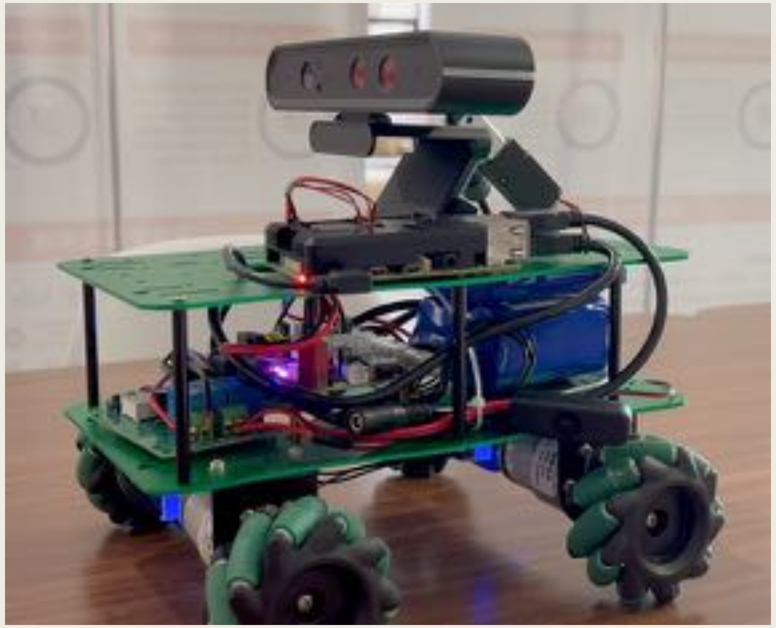
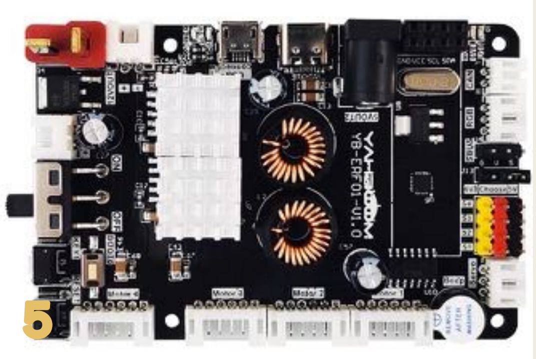


Figure 2: Robot Car

- **Remote Server Subsystem:** Do the visualization and the 3D-reconstruction
- **Communication Subsystem:** Works as a communication bridge between the server and the car
- **Robot Car Subsystem:** A car platform that supports omnidirectional movement controlled by a joystick, holds an RGBD camera to gather information

Components

- 1. Car Platform
- 2. RGBD Camera
- 3. Raspberry Pi
- 4. Xbox Joystick
- 5. STM32-based Control Board
- 6. Linear Actuator
- 7. Customized PCB
- 8. Server Computer



Methodology: Remote Control

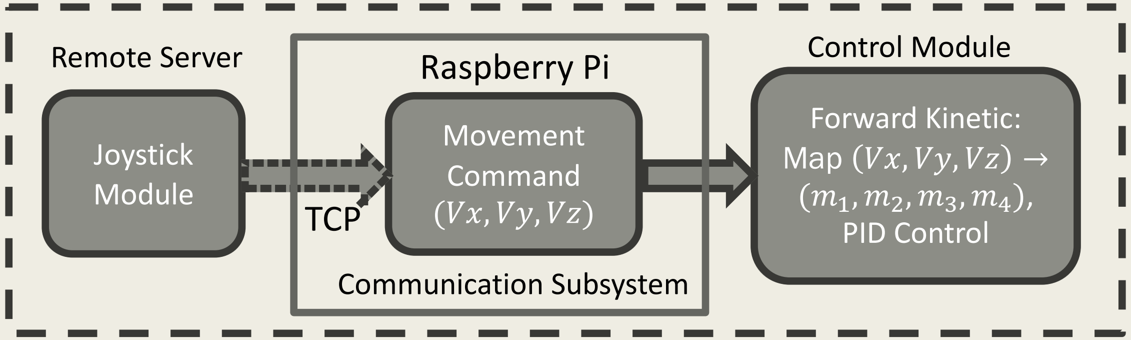


Figure 3: Remote Control Workflow

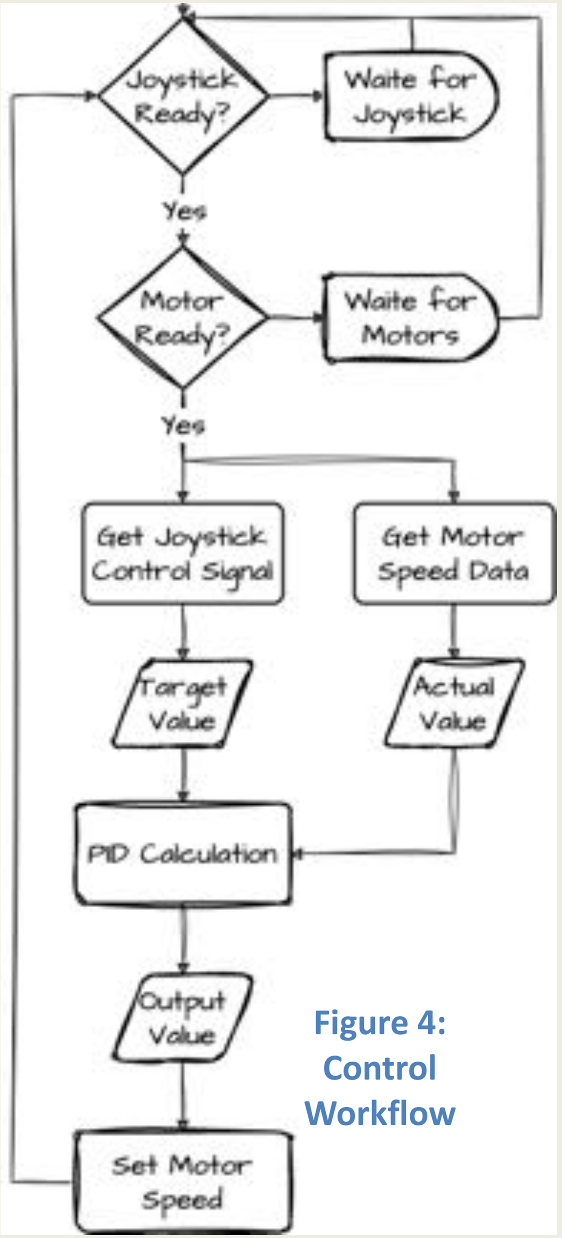


Figure 4: Control Workflow

Methodology: Remote Control

McNamee Wheels and Motors

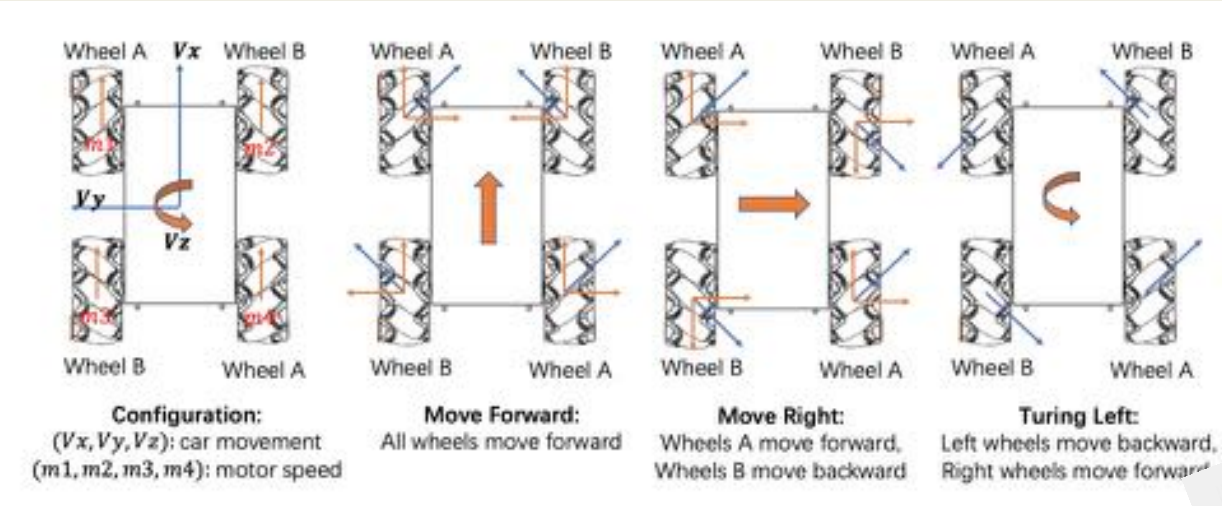


Figure 5: Configuration and Simple Kinetic Analysis of McNamee Wheels

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix} = \frac{1}{r}\begin{bmatrix}1 & -1 & -(l_x+l_y)\\ 1 & 1 & (l_x+l_y)\\ 1 & 1 & -(l_x+l_y)\\ 1 & -1 & (l_x+l_y)\end{bmatrix}\begin{bmatrix}v_x\\v_y\\\omega_z\end{bmatrix}.$$

$$\begin{cases}\omega_1 = \frac{1}{r}(v_x - v_y - (l_x+l_y)\omega),\\ \omega_2 = \frac{1}{r}(v_x + v_y + (l_x+l_y)\omega),\\ \omega_3 = \frac{1}{r}(v_x + v_y - (l_x+l_y)\omega),\\ \omega_4 = \frac{1}{r}(v_x - v_y + (l_x+l_y)\omega).\end{cases}$$

Equation 1: Forward Kinetic

Joystick Module

Name	Function	Name	Function
Left Stick X-axis	Move Left/Right	Left Stick Y-axis	Move Forward/Backward
Left Trigger	Turn Left	Right Trigger	Turn Right
Left Button	Decrease Max Turning Speed	Right Button	Increase Max Moving Speed
Start Button	Emits a beep	Back Button	Stop Moving

Table 1: Joystick Function Map

$$\begin{bmatrix}v_x\\v_y\\\omega_z\end{bmatrix} = \frac{r}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ -1 & 1 & 1 & -1\\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)}\end{bmatrix}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}$$

Longitudinal Velocity:

$$v_x(t) = (\omega_1 + \omega_2 + \omega_3 + \omega_4) \cdot \frac{r}{4}$$

Transversal Velocity:

$$v_y(t) = (-\omega_1 + \omega_2 + \omega_3 - \omega_4) \cdot \frac{r}{4}$$

Angular velocity:

$$\omega_z(t) = (-\omega_1 + \omega_2 - \omega_3 + \omega_4) \cdot \frac{r}{4(l_x+l_y)}$$

Equation 2: Backward Kinetic

Methodology: Remote Control

McNamee Wheels and Motors

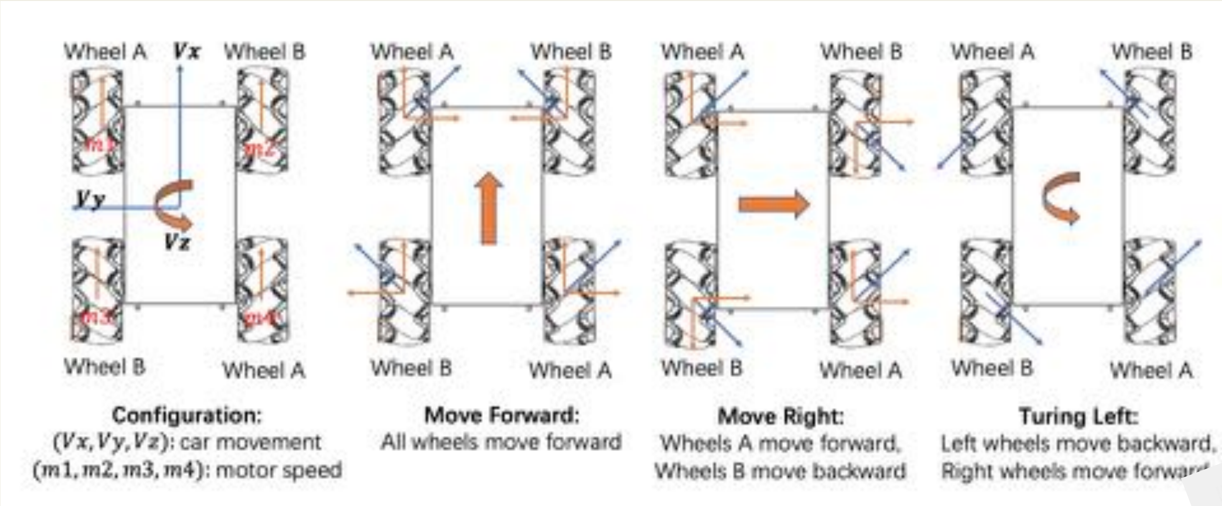


Figure 5: Configuration and Simple Kinetic Analysis of McNamee Wheels

Joystick Module

Name	Function	Name	Function
Left Stick X-axis	Move Left/Right	Left Stick Y-axis	Move Forward/Backward
Left Trigger	Turn Left	Right Trigger	Turn Right
Left Button	Decrease Max Turning Speed	Right Button	Increase Max Moving Speed
Start Button	Emits a beep	Back Button	Stop Moving

Table 1: Joystick Function Map

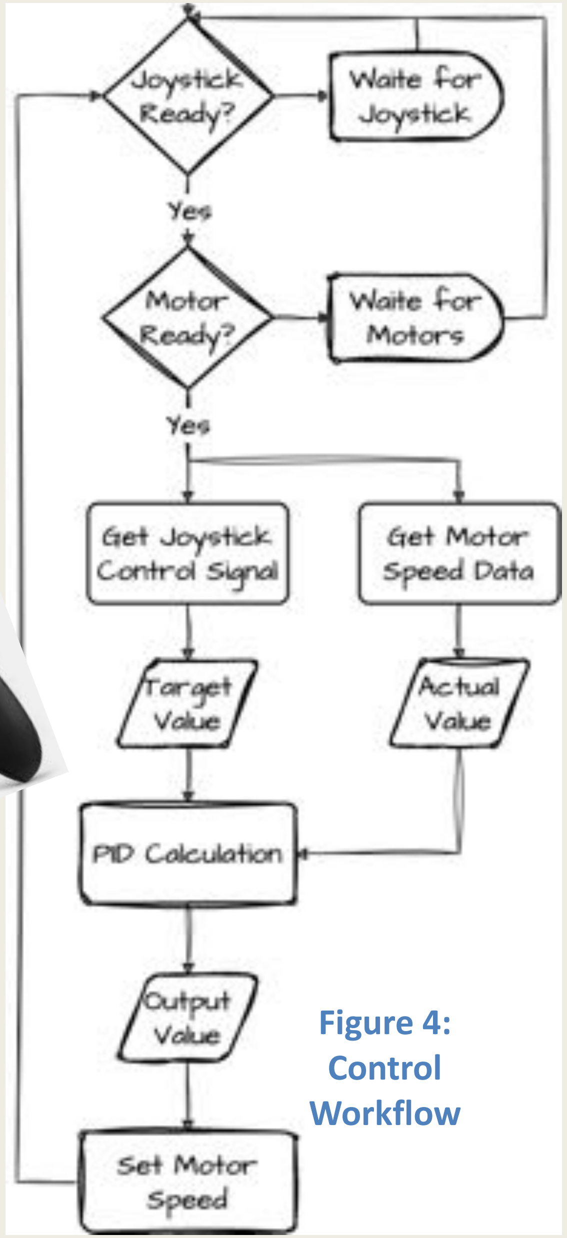


Figure 4: Control Workflow

Methodology: Image Transmission

- Transmit RGB and depth images through WIFI
- Images are compressed using JPEG image compression algorithm on the Raspberry Pi before transmission to save bandwidth.
- The compression rate can reach nearly **10x**, resulting in a significant bandwidth reduction.

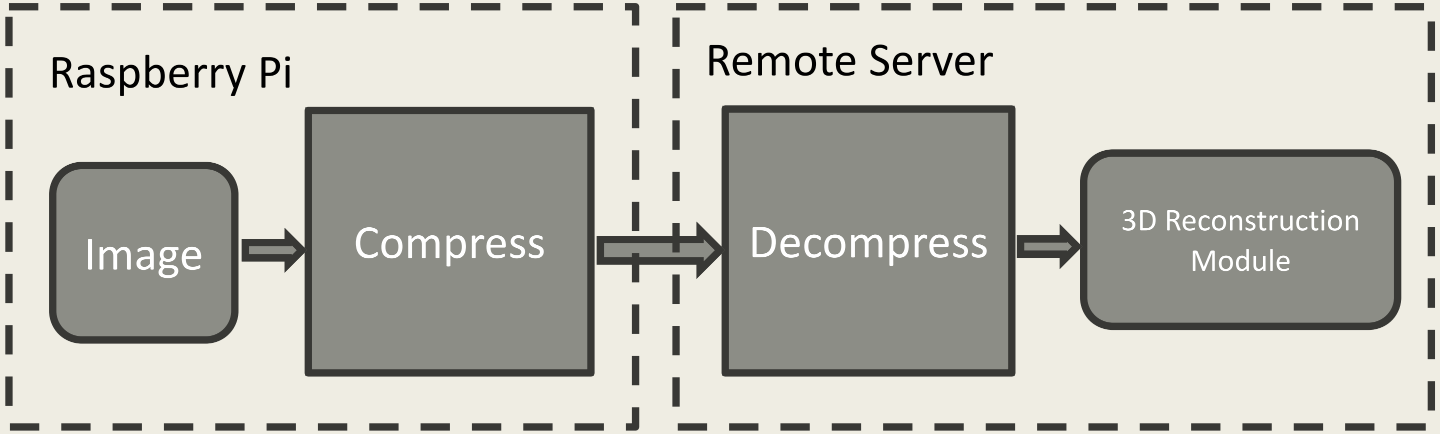


Figure 6: Image Compression Workflow

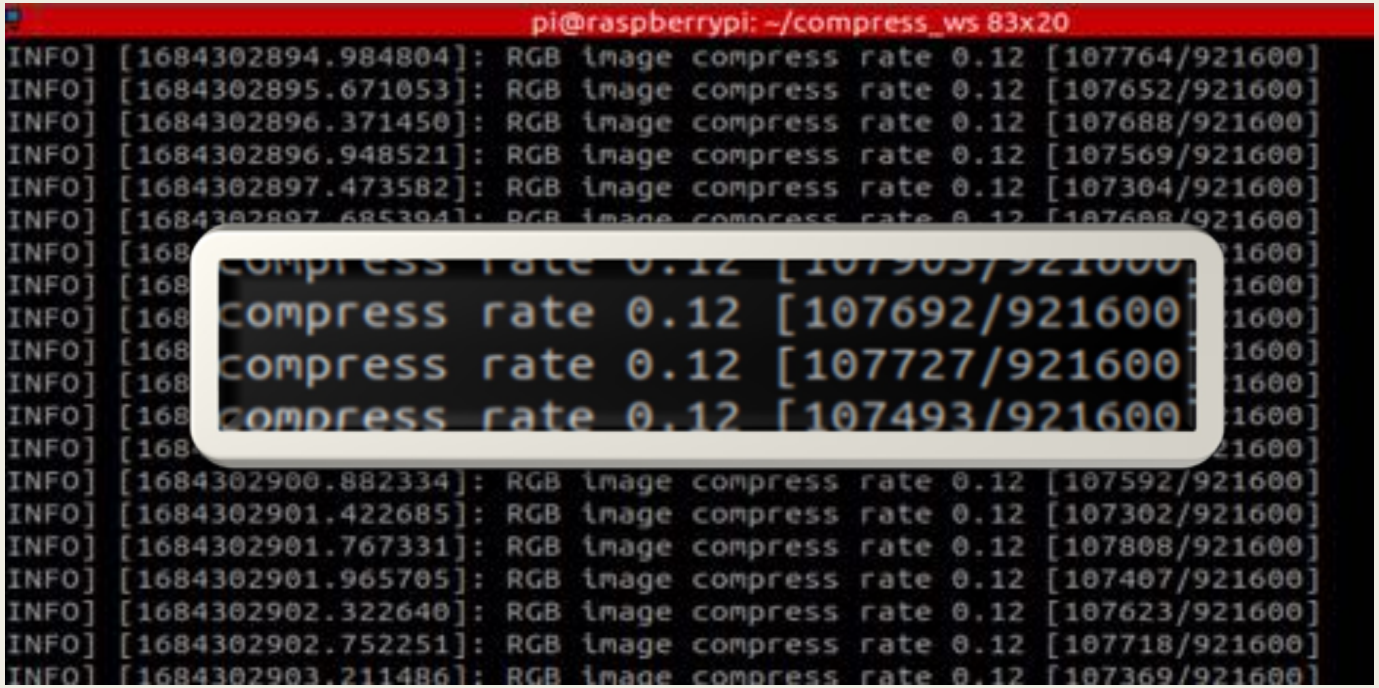


Figure 7: Image Compression Rate

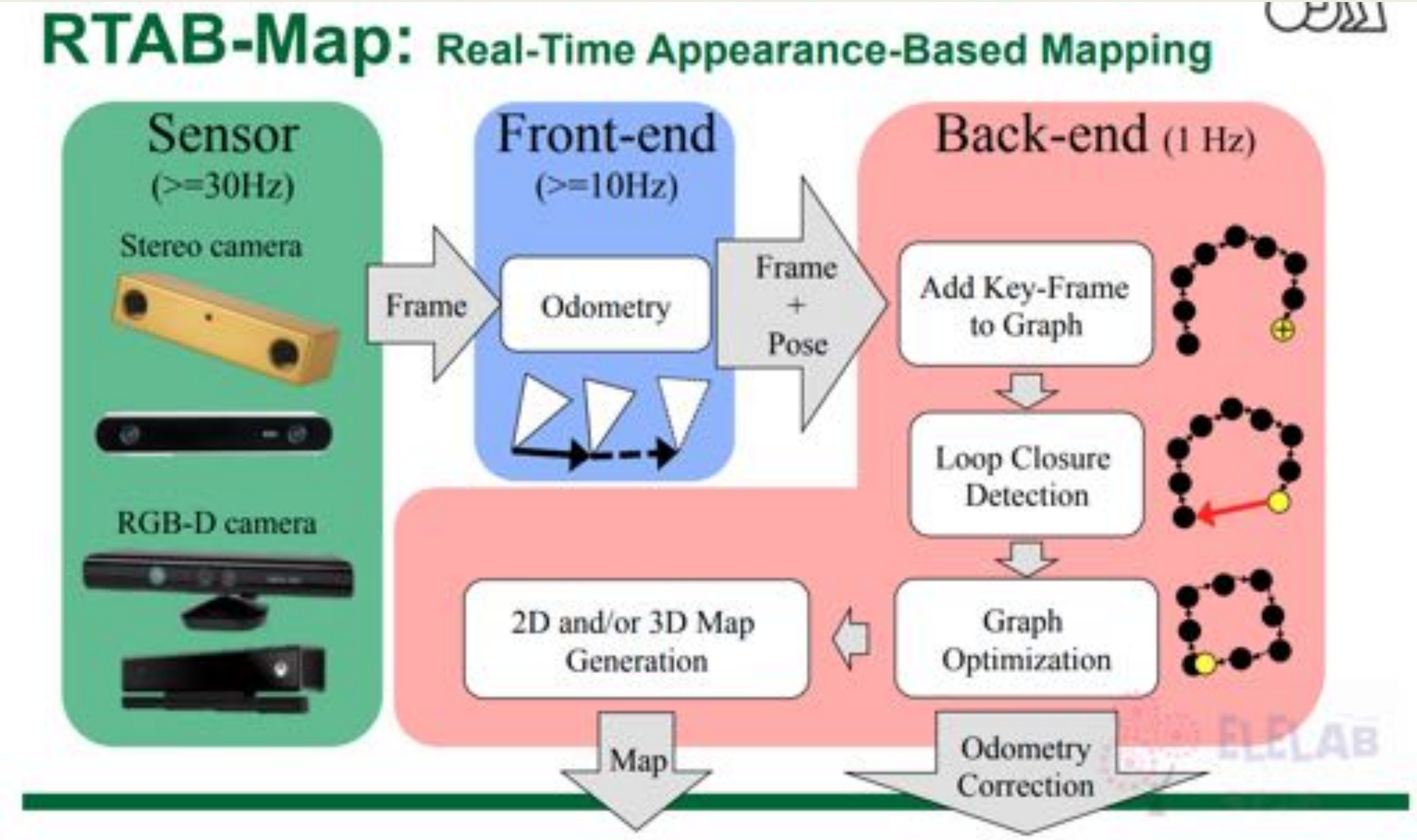


Figure 8: Intro to RTAB-Map

3D Reconstruction Algorithm

- Image Denoising with Mean Filter
- Image Sampling
- Feature Matching
- Trajectory Calculation
- Closure Detection
- Calibration
- Post-Processing

Figure 9: Reconstruction Result

USER INTERFACE

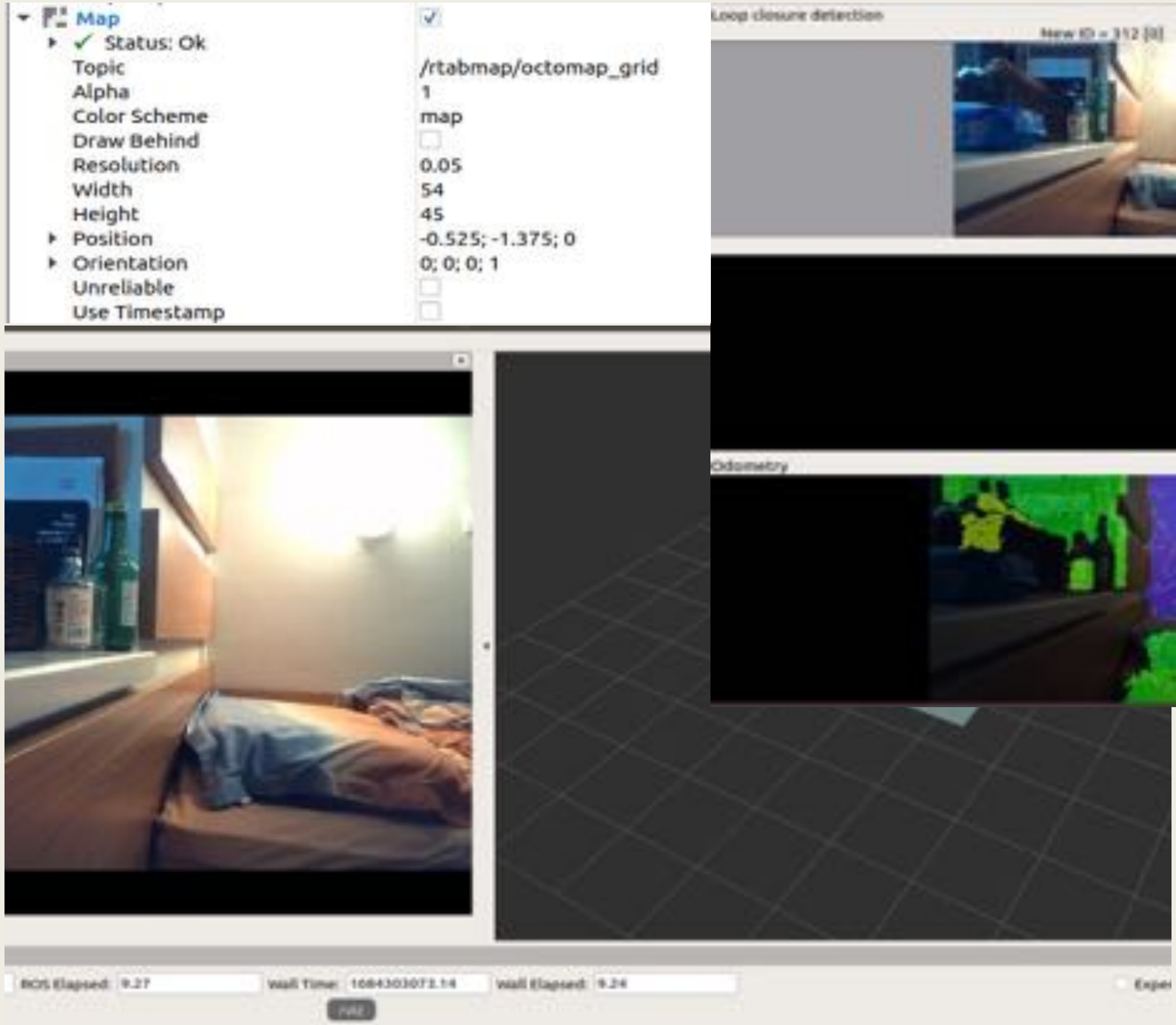


Figure 10: View from Rviz

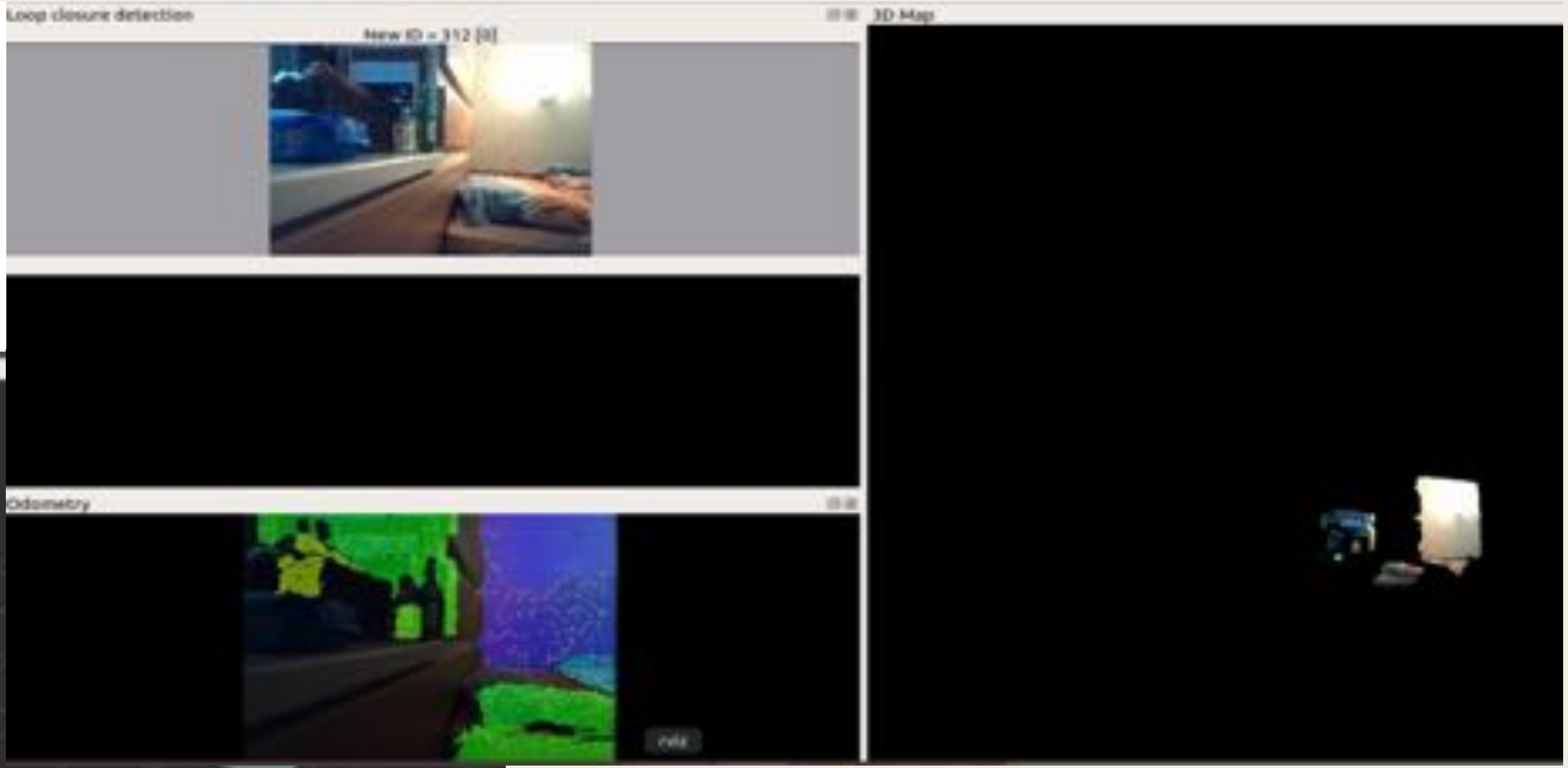


Figure 11: View from R-tab-Map

Mechanical Structure

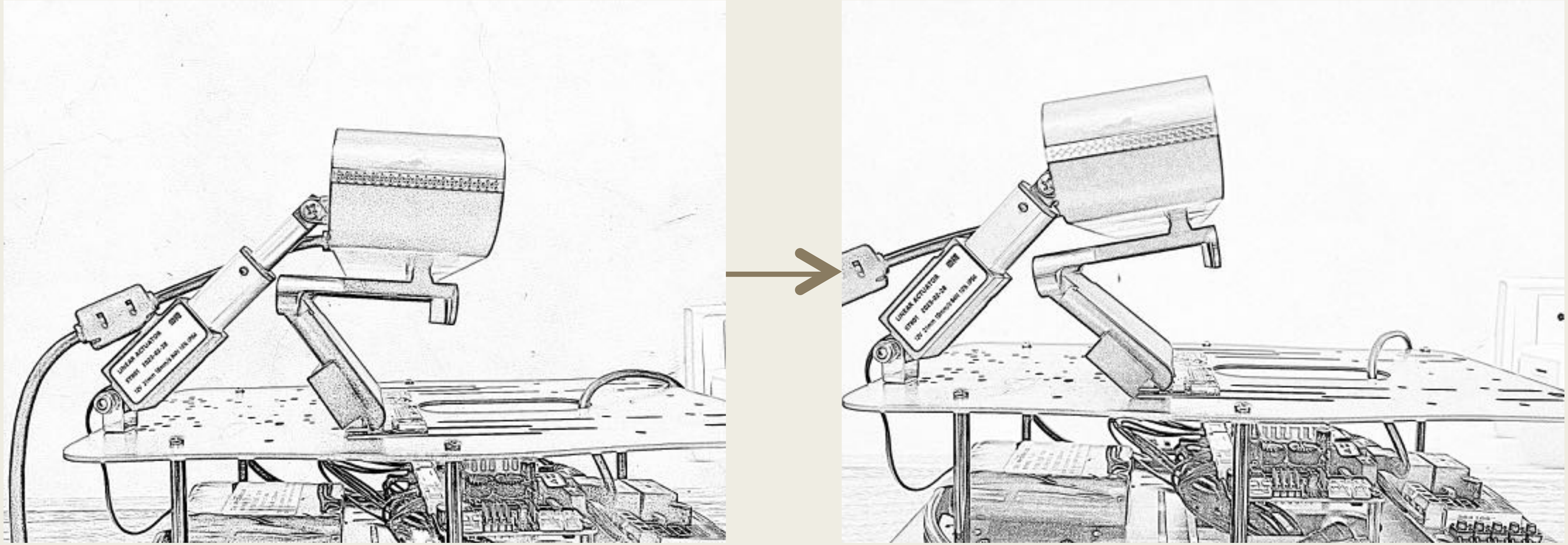
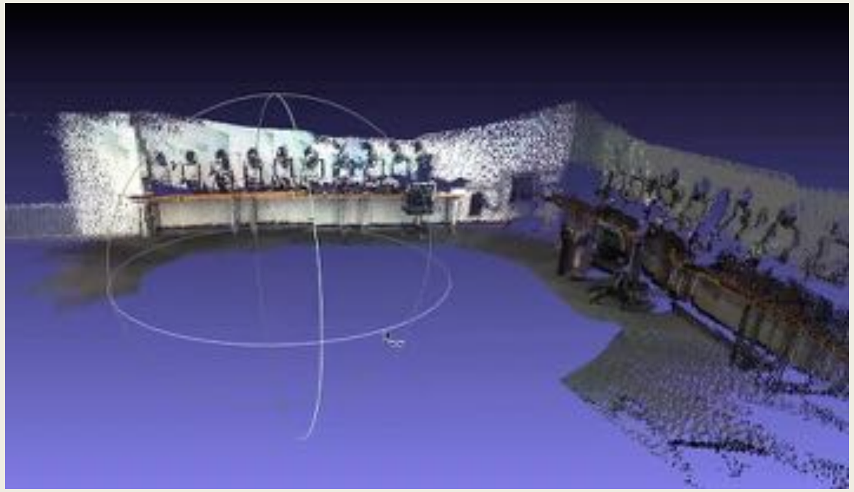
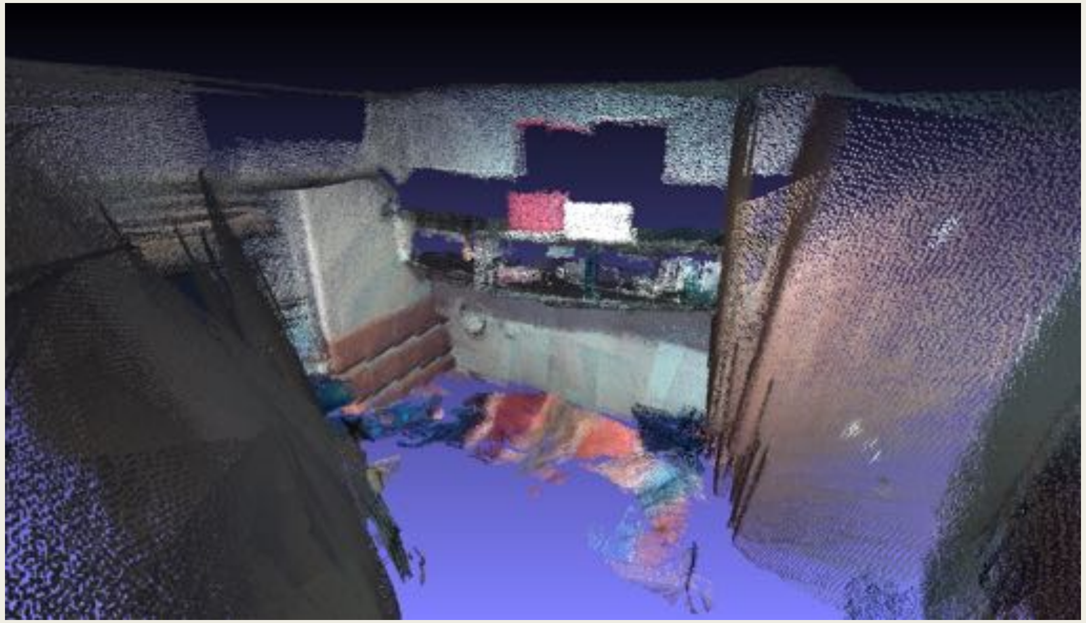
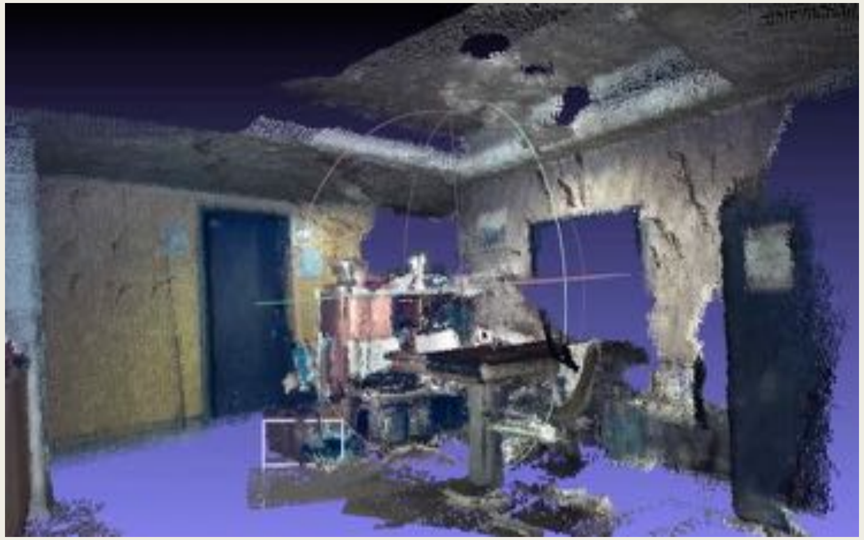


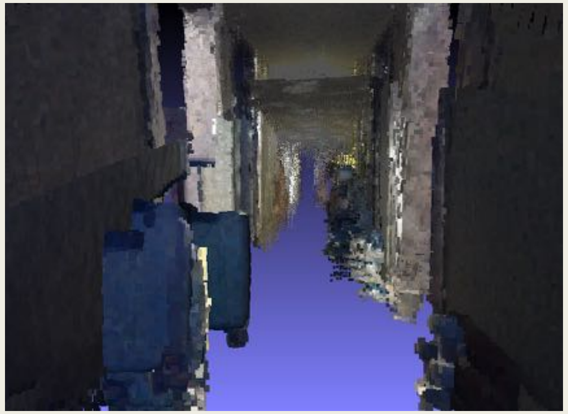
Figure 10: Linear Actuator

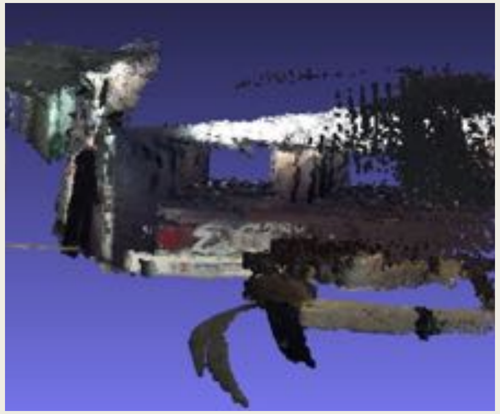
Result: Impressive Reconstruction











Result: Strengths

Smooth Control: user can control the car to move in all directions with low delay and the car will response immediately

Bandwidth Saving: Even though we require high resolution images to perform a great result, our compress step save the bandwidth to 25% as original

Real Time Construction: our reconstruction result will be real time with little delay and the final result can be refined with extra time.

User Friendly Interface: our interface shows the current scene of the camera and the reconstruction result. The user can switch from one mode to the other simply press a button.

Result: Weaknesses

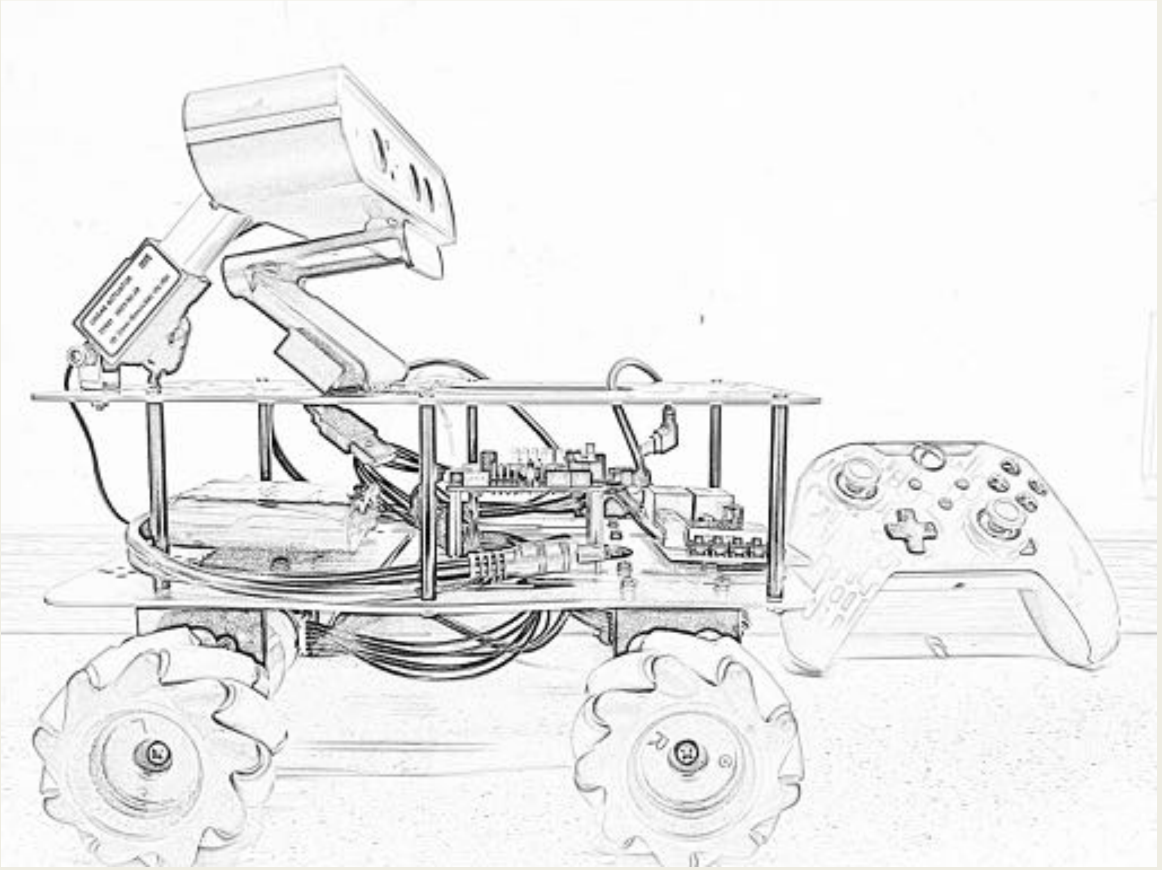
<u>High-quality Network Required</u>: since the reconstruction and the control modules are implemented on the remote server, it is crucial to have a good network condition to work.

<u>Plain Background Forbidden</u>: our algorithm is based on the traditional method, and the features are extracted from the color gradient. In this case, the algorithm cannot work with plain background.

<u>Holes In The Point Cloud</u>: Due to height of the camera, the reconstruction may contain some holes since the camera is blocked by the objects. The wall behind the table and the 3d printers are the vivid examples.

Conclusion

- In conclusion, our senior design project has successfully demonstrated the feasibility and efficiency of utilizing an RGBD camera for 3D reconstruction in a robotic car context.
- The methodologies employed, from image denoising and sampling, feature matching and trajectory calculation, to closure detection and calibration, have proved instrumental in achieving high-quality results.
- The precision and accuracy of our 3D reconstruction module have been evidenced in real-world environments, as seen in the lab scene we reconstructed, despite some limitations due to the camera's fixed position.
- These results underscore the robustness and effectiveness of our system, and its potential in fields requiring detailed environmental understanding.



Remote Robot Car Control System with RGBD Camera for 3D Reconstruction

Thank You for Listening!

Any Questions?

ECE445 Senior Design
SP23 ZJUI TEAM#21