Nathan, Rick, Steven, Willy
Team Members

Nathan Bornfreund
Rick Franc
Steven Landis
Willy Tu
Background

- TrueVision Alcon (TVA) help surgeons during operation by providing 3D surgical visualization
- Captured with a robotic stereoscopic digital microscope
- Universal Robotics 5-kg payload (UR5) robotic arm
Overview

- 6 degrees of freedom
- Operation Modes
  - Freedrive
  - Remote control
  - Automatic mode
- The arm is unaware of surroundings
- SafeVision prevents collisions
Application
System Requirements

- Prevent object collision
- Safely stop the arm which can move up to 0.8 m/s
- Move predictably
- Use parts approved for medical environments
- Weigh less than 1 kg
Proposed Solution

- Use distance sensors to detect objects in direction of movement
- Use a model of the arm to prevent self-collision
- Ensure safe conditions before resuming movement
Hardware
Hardware Schematic

Optical Sensor 1

I2C_1

Optical Sensor 2

I2C_2

Optical Sensor 3

Optical Sensor 4

Microcontroller

I2C_4

Microcontroller
Processor

STM32F767ZI

- High-performance
- Arm Cortex M7 processor
- 4 x I2C buses
- UART communication
- ST Link Debugger
Sensors

VL53L0X Time of Flight Sensor

- Takes 20 ms to get measurement
- Accurate from 2.5 cm to 80 cm
- 25 degree FoV
- Communicates over I2C
- Shielded wire to minimize crosstalk
Sensor Mounts
Sensor Placement
PCB Design

- 4-layer PCB
- 6 x 7 in
- Two different power sources
  - On-chip components
  - External distance sensors
Microcontroller Software Workflow

- Initialize Peripherals
- Setup Sensors
- Read Sensor Data (I2C)
- Transmit Data (UART)

15-25 ms delay
Software
Object Detection Workflow

- Sensor Data
  - UART
- Control System
  - Update Arm Status
  - RTDE
- UR5 Arm
  - Update Arm Position
- Serial Port Communication
- Object Detection
  - Update Distance Data
  - Visualization Detection
- Robotic Arm Visualization
- Android App
  - Remote
Universal Robotics Arm

- Real-time data exchange (RTDE)
- URScript Programming Language
- Polyscope UR Simulator
Object Detection
Arm-Arm collision

- Model the arm as a series of rectangular prisms
- If any of the prisms intersect, the arm is about to collide with itself
Remote Control App

- Connect to Windows server with TCP client
- Visualization
- Control arm angles
Challenges

- Enabling sensors in motion
  - Ignore self-detection
- Minimizing latency
  - 0.5 second response time
  - Detect, process, and decelerate
- COVID-19 response
  - Situation testing
  - Android app
Acknowledgements

We would like to thank Yuepei Hu and Patrick Terry from Alcon, Professor Yogananda Isukapalli, and our TAs Kyle Douglas and Aditya Wadaskar for making this project possible.