

# **Gesture Controls for the Raven Robot**

CS 446 Computer Integrated Surgery II

Project Proposal, February 22, 2013

Alan Chancellor and Kristine Sarnlertsophon

Mentors: Kelleher Guerin and Anton Deguet

## **Project Summary**

We will write software to control the Raven Surgical Robot with hand gestures. We will do this by integrating the 3Gear System, CISST libraries, the Robot Operating System (ROS), and the Raven itself. This will not only provide a new input to the Raven robot (which currently can only be controlled using Phantom Omnis, a specific piece of hardware) but also join many key libraries used in working with surgical robots.

## **Motivation and Significance**

The Raven Robot is a surgical robot with open-source software, created by the University of Washington in Seattle. It has been sent to research labs across the country as a platform for researchers to experiment with surgical robots. We aim to increase its usability by integrating the CISST libraries and a gesture control input; in doing so, we will create interfaces for:

1. 3Gear to CISST communication,
2. CISST library to ROS, and
3. ROS to the Raven robot (or Raven simulator).

We are using the Raven Robot as a vehicle to integrate these software libraries, and the goal of our project is to demonstrate this integration by controlling the Raven Robot using hand gestures. The integration of these systems will be of use to researchers working with surgical robots, even if they don't use the Raven robot. By enabling the 3Gear device to communicate using the CISST library, we open the doors for this more natural form of input to be used in other surgical robot systems. In linking the CISST library to ROS, we allow the users of ROS (there are many) access to a well-developed library specifically for the development of computer assisted intervention systems. Finally, we are refining the Raven Robot's software to better integrate with ROS so that other researchers may further develop using the Raven.

# Background

## 3Gear

The 3Gear system is a commercial product, currently in beta, which utilizes two Kinects (for their 3D cameras) to track a user's hand gestures, accurate to millimeters. 3Gear provides Java and C++ libraries which allow us to write our own hand-tracking applications. Their software passes messages about the position and orientation of each hand, which we can use to control the hands on the Raven. 3Gear's own software relies on user-calibrated poses to track the hands; since they have not developed tracking for poses outside of their database, we may need to see what effect this limitation has on our own system.

## CISST Library

The CISST package is an open-source collection of libraries designed to be used in computer assisted intervention systems. We will be extensively using the CISST Multitask library, which provides the component-based framework for the CISST package. Objects in the CISST Multitask framework include "provided interfaces" and "required interfaces," which allow for communication between different components in a client-server manner. This allows for our code to be far more reusable; ideally, the CISST libraries we write will be able to plug in to any input device with an appropriate SAW/CISST Multitask wrapper (such as the one we are writing for the 3Gear) and transmit the appropriate data to ROS.

## ROS

The Robot Operating System is an open-source operating system for robots. It provides hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. ROS is a distributed framework of modular "nodes," which can be put together to form an entire system. As such, ROS code is extremely reusable; we aim to take advantage of the nature of ROS and CISST Multitask to make the software we write for the 3Gear to Raven system generalizable for many inputs and robots.

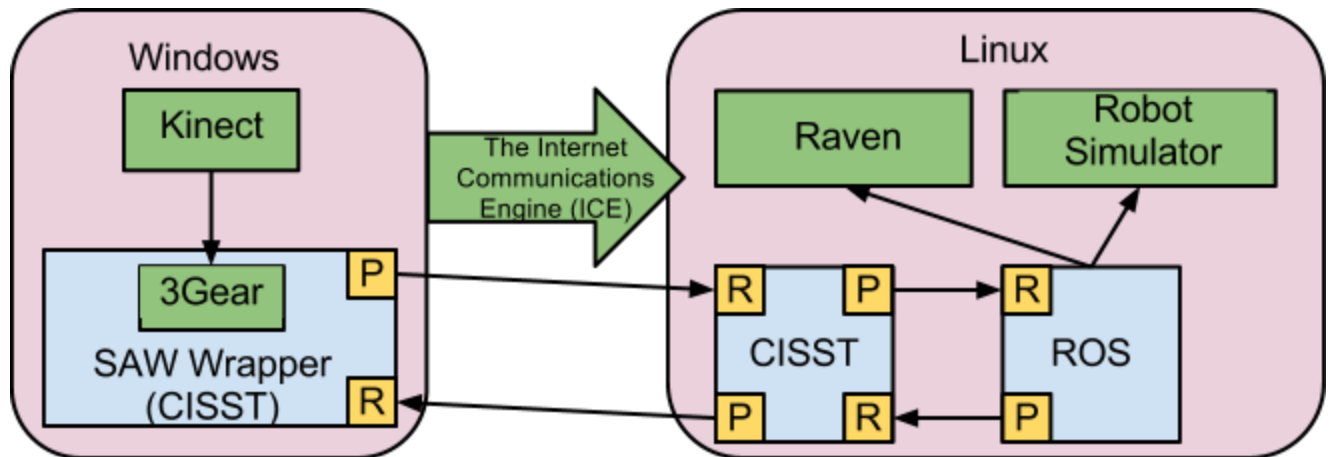
## Raven Robot

The Raven is a robotic surgery research system that uses open-source software based on ROS. It was developed by the BioRobotics Laboratory in the University of Washington in Seattle, and it has been sent to multiple research laboratories to enable further research in tele-operative and minimally invasive surgery. Although it has not yet been approved by the FDA, research laboratories have ambitious plans for the Raven, including operating on a beating heart (moving in sync with the heartbeats) and having the robot perform autonomously by imitating surgeons.

# Technical Approach

The components will be made to interface with each other via a component-based approach. This means writing a SAW (Surgical Assist Workstation; CISST package) wrapper for 3Gear, and writing the required and provided interfaces between CISST and ROS. It will also be necessary to make

CISST communicate between an iteration running on a Microsoft Windows machine, which runs the 3Gear input system, and an Ubuntu machine, which runs the ROS and the Raven or robot simulator; communication between the two computers will be achieved using ICE, the Internet Communications Engine from ZeroC.



## Deliverables

### Minimum: (Expected by April 19)

- SAW Wrapper for 3Gear + documentation
- CISST to ROS interface + documentation
- Simple frames moving in Gazebo (demonstration)

### Expected: (Expected by April 26)

- 3Gear to CISST interface + documentation
- CISST to ROS interface + documentation
- ROS to Raven Simulator interface + documentation
- Show control of Raven Simulator using gestures (demonstration)

### Maximum: (Expected by May 3)

- ROS to Raven robot interface (all libraries + documentation)
- Show control of Raven Robot using gestures (demonstration)

## Milestones

1. Milestone name: Design Specifications
  - Planned Date: March 11

- Expected Date: March 11
- Status: In Progress
- 2. Milestone name: 3Gear to CISST code + documentation
  - Planned Date: April 1
  - Expected Date: April 1
  - Status: In Progress
- 3. Milestone name: CISST to ROS code + documentation
  - Planned Date: April 1
  - Expected Date: April 1
  - Status: In Progress
- 4. Milestone name: ROS to Raven Simulator code + documentation
  - Planned Date: April 19
  - Expected Date: April 19
  - Status: Planned
- 5. Milestone name: 3Gear to Raven Simulator Demonstration
  - Planned Date: April 26
  - Expected Date: April 26
  - Status: Planned
- 6. Milestone name: ROS to Raven Robot code + documentation
  - Planned Date: April 29
  - Expected Date: April 29
  - Status: Planned
- 7. Milestone name: 3Gear to Raven Simulator Demonstration
  - Planned Date: May 3
  - Expected Date: May 3
  - Status: Planned

## Dependencies

### Access to 3Gear Computer

- Resolution Plan: Get J-Card access, ask Kell for access
- Resolve by: February 15
- Resolved: Yes
- Fallback plan: N/A

### Learn to build CISST

- Resolution Plan: Meet with Anton to help us
- Resolve by: February 22
- Resolved: Yes
- Fallback plan: Find another helpful person to teach us CMake (or learn it through documentation)

### Access to Linux Machine for ROS and Raven Simulator

- Resolution Plan: Kell finds us a Linux machine to work with
- Resolve by: February 22
- Resolved: No
- Fallback plan: Work on a virtual machine or on our laptops

#### **Software Design Approval from Mentors**

- Resolution Plan: Meet with Kell and Anton to review and approve software plan
- Resolve by: March 6
- Resolved: No
- Fallback plan: Meet with Prof. Taylor to review software (will likely cause delay)

#### **Networking between 3Gear (Windows) machine and ROS/Raven (Linux) Machine**

- Resolution Plan: Ask mentors for help
- Resolve by: March 25
- Resolved: No
- Fallback plan: Learn to network computers?

#### **CAD models and/or actual Raven Simulator**

- Resolution Plan: These should be available through the Raven community
- Resolve by: March 15
- Resolved: No
- Fallback plan: Contact other research groups/universities for access to their Raven Simulator, or use the Gazebo simulator

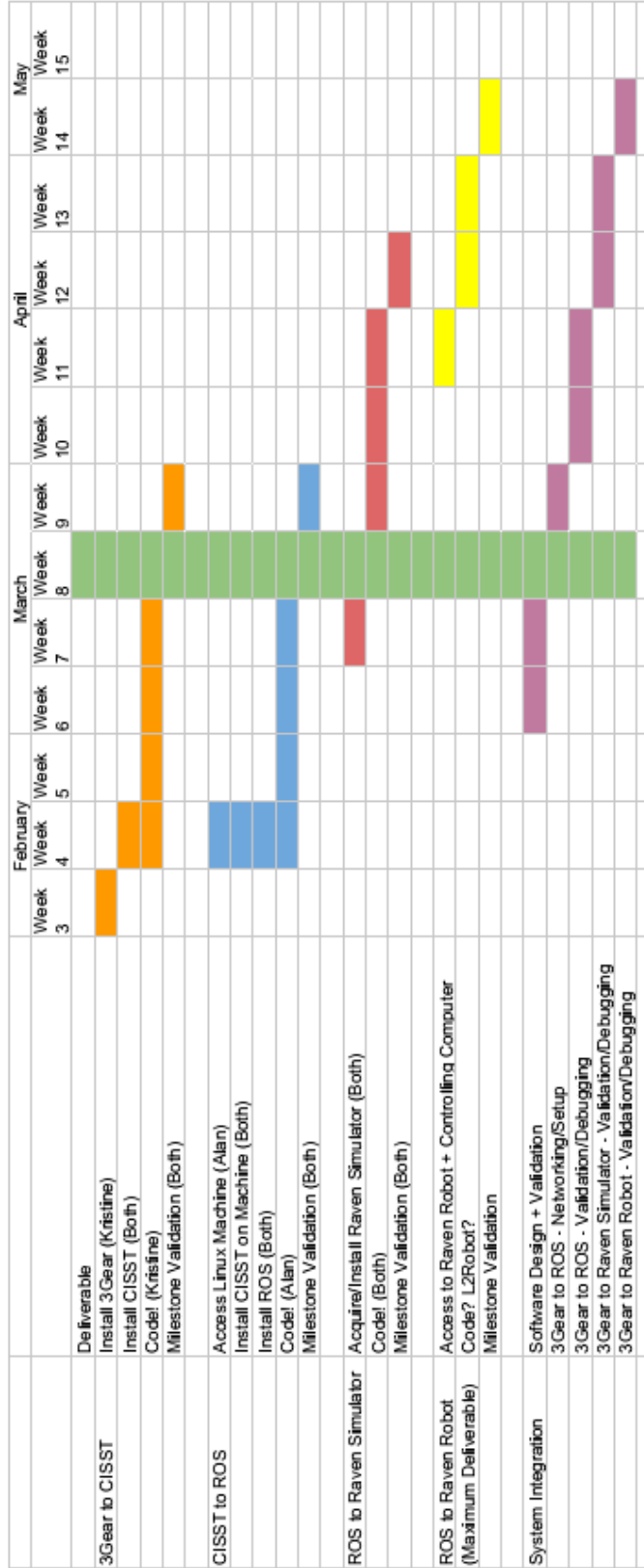
#### **Access to Raven Robot + Control Computer**

- Resolution Plan: Ask Kell for access
- Resolve by: April 15
- Resolved: No
- Fallback plan: Maximum Deliverable is not achieved.

## **Management Plan**

We will check in with our mentors weekly on Wednesdays at 1pm in order to ensure that our deliverables and dependencies are on track. We will also meet afterwards (sans mentors) in order to discuss any issues or upcoming deadlines, as well as update our project wiki page. Outside of these meetings, we each plan on working approximately 6-10 hours per week on the project.

# Project Timeline



## References

M.J.H. Lum, J. Rosen, T.S. Lendvay, M.N. Sinanan, B. Hannaford, 'Effect of Time Delay on TeleSurgical Performance,' IEEE International Conference on Robotics and Automation (ICRA), 2009.

M.J.H Lum, J. Rosen, H. King, D.C.W. Friedman, G. Donlin, G. Sankaranarayanan, B. Harnett, L. Huffnam, C. Doarn, T. Broderick, B. Hannaford, 'Telesurgery Via Unmanned Aerial Vehicle (UAV) with a Field Deployable Surgical Robot,' Proceedings, Medicine Meets Virtual Reality (MMVR), Long Beach, CA, 2007.

<https://trac.lcsr.jhu.edu/cisst>

Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

<http://www.threegear.com/technology.html>

<http://www.ros.org/wiki/>