An Autonomous Camera System using the da Vinci Research Kit

Shahab Eslamian, Luke A. Reisner, Brady W. King,
and Abhilash K. Pandya*, Member, IEEE

*Associate Professor, Electrical and Computer Engineering

Computer-Assisted Robot-Enhanced Systems (CARES) Lab

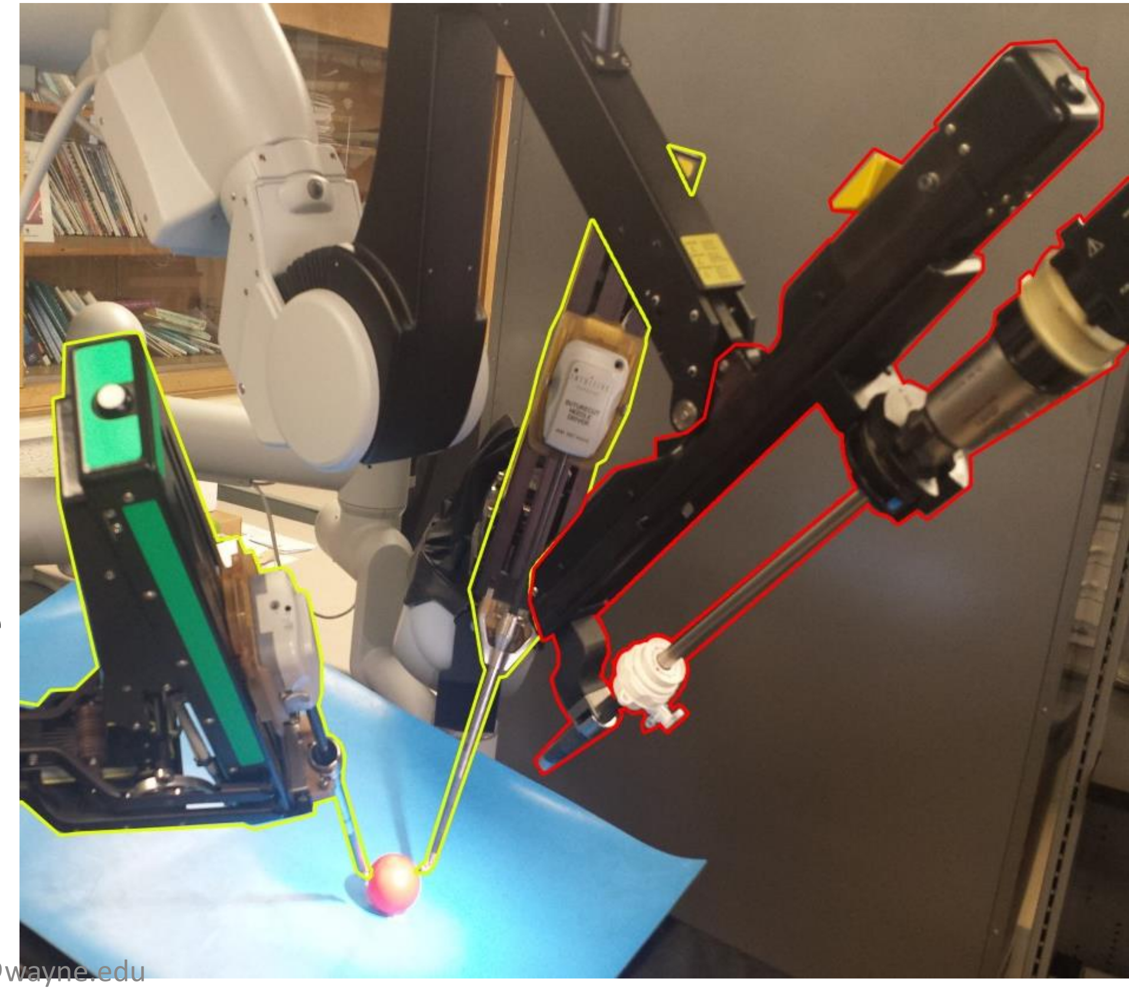
Wayne State University, Detroit, MI

Introduction

- Current surgical camera control (using a human assistant or teleoperated robot) can be inefficient, error prone, and costly
  - We've observed camera movement interrupting the flow of surgery up to 100 times/hour
  - Suboptimal camera views are sometimes selected to avoid interruption
  - Tool(s) sometimes leave the field of view, posing a risk to the patient
- We have created a test platform using the da Vinci Research Kit (DVRK) that implements basic autonomous control of the camera

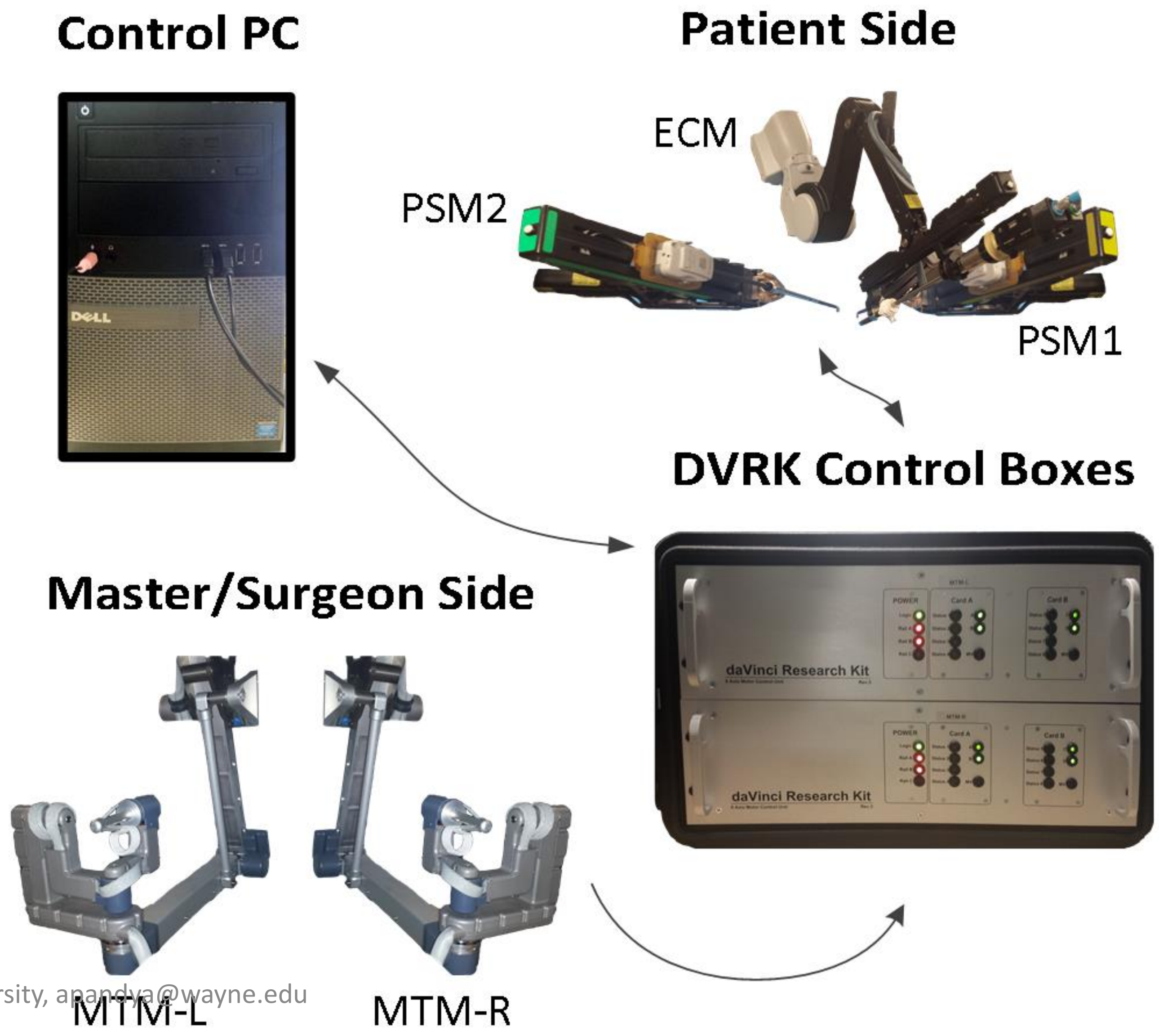
Overview of Methods

- Hardware setup
  - da Vinci Research Kit software and FPGA boards
  - PID parameter calibration
  - Robot arm co-registration
  - Camera calibration parameters
- Software development
  - Reimplemented the teleoperation module
  - AutoCamera control algorithms
- Basic accuracy and usability testing



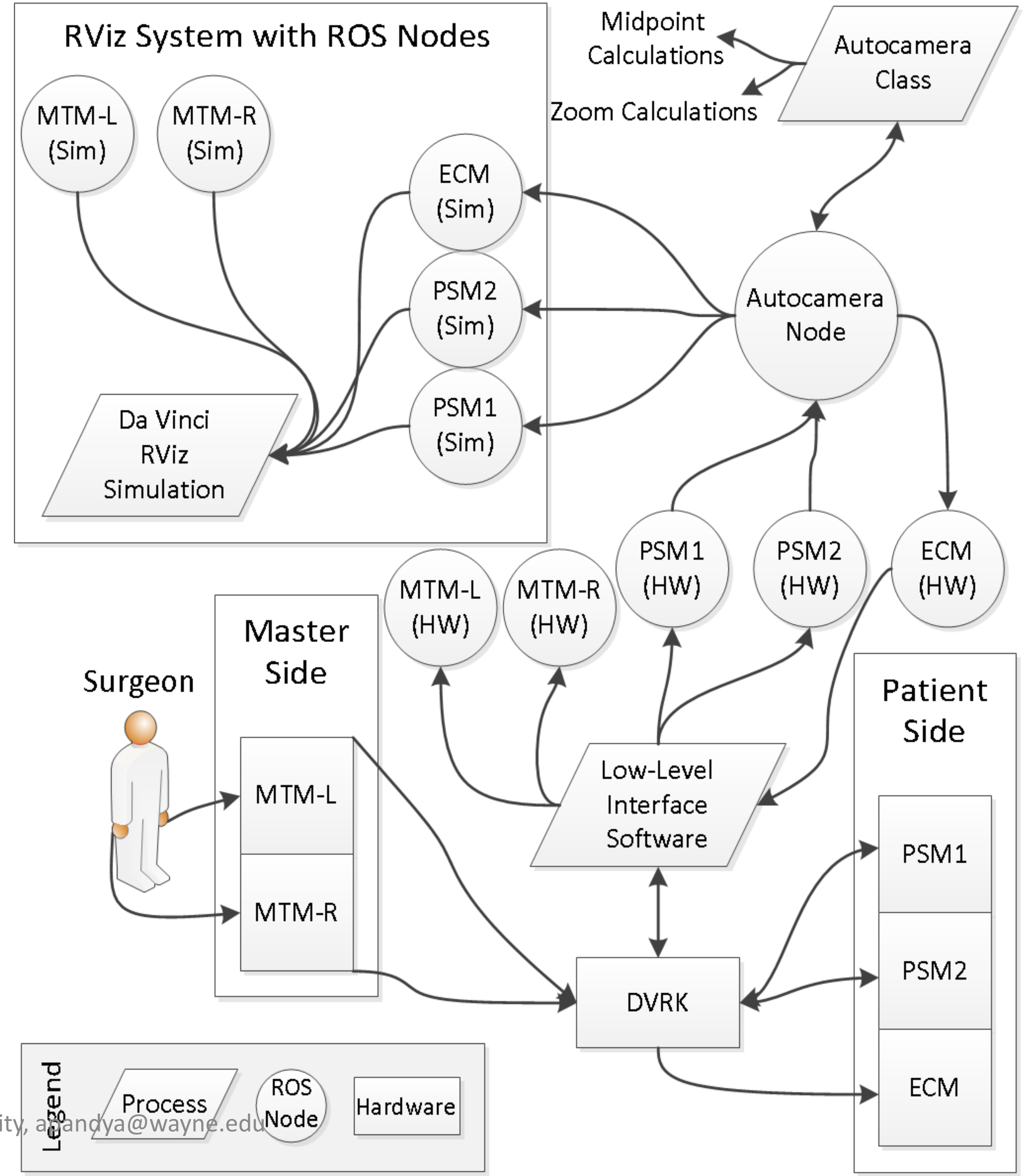
System Hardware

- Full da Vinci Standard Surgical System
- Da Vinci Research Kit (DVRK) control boxes
- PC with Ubuntu and the Robot Operating System (ROS) framework

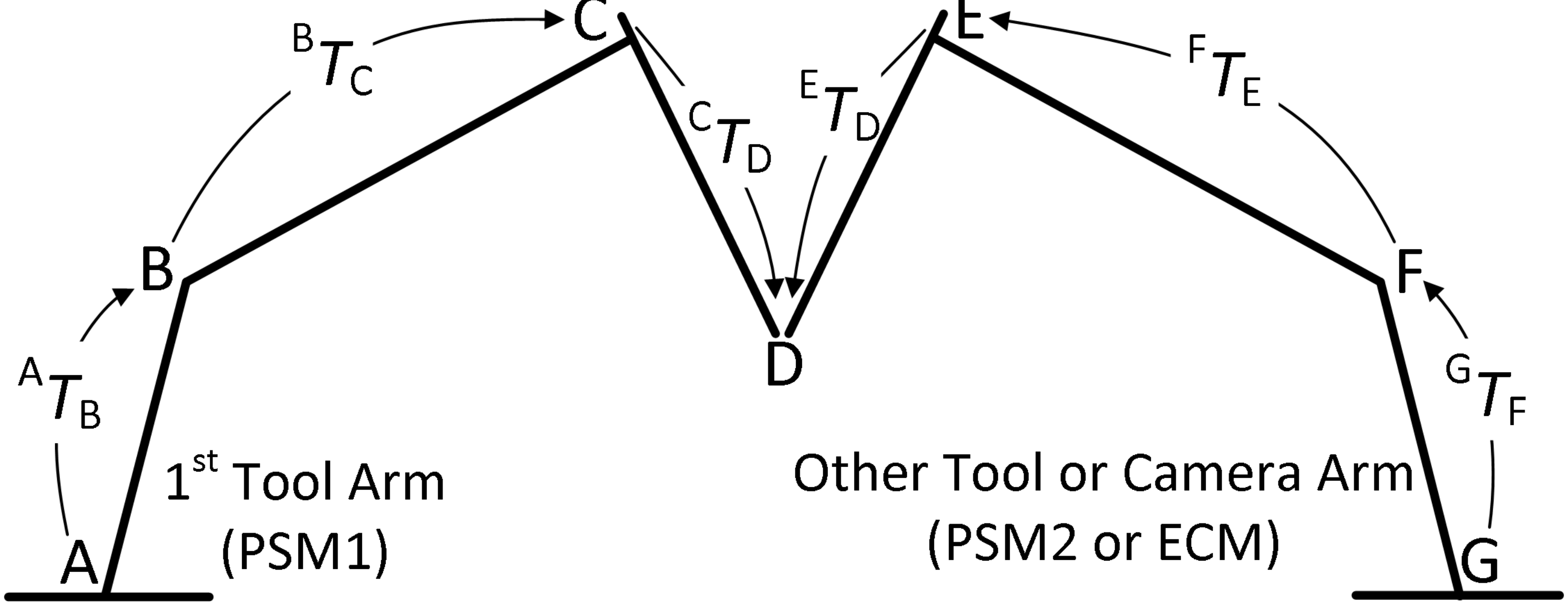


System Architecture

- Data flows through ROS nodes
- The DVRK reads the poses of the robot arms as the surgeon moves the hand controllers
- AutoCamera software moves the camera arm (ECM)
- Robot's state is reflected in a simulation environment (RViz)



da Vinci Robot Arm Co-registration



Other arm's base relative to PSM1 base:

$$^{A}T_{G} = {}^{A}T_{B}\,{}^{B}T_{C}\,{}^{C}T_{D}\left({}^{E}T_{D}\right)^{-1}\left({}^{F}T_{E}\right)^{-1}\left({}^{G}T_{F}\right)^{-1}$$

- Tips of arms touched together in ~6–10 different configurations
- Error is computed as the calculated distance between the tips (should be ~0)
- Error minimized by optimizing a 3D transformation between the arms' bases

Stereo Camera Calibration

- The zoom control algorithm relies on projecting points in the 3D view to 2D pixel coordinates
  - Camera calibration parameters needed to perform this projection
- We used the camera_calibration package in ROS
- The identified camera calibration parameters include:
  - Focal length
  - Field of view
  - Distortion parameters
  - Projection matrix
  - Rectification matrix

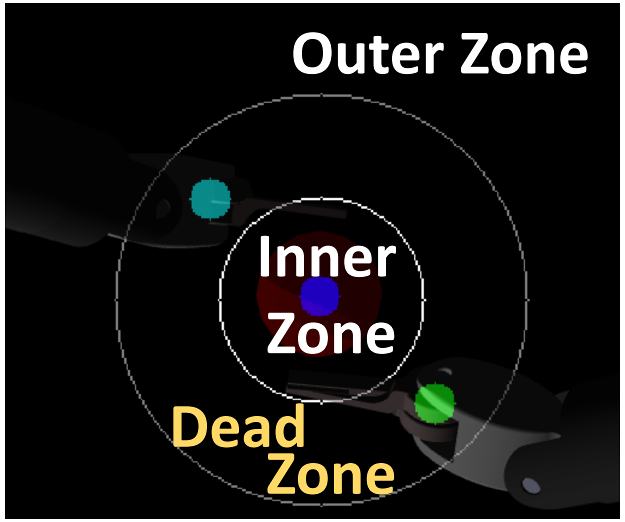
AutoCamera Midpoint Tracking Algorithm

1. Compute the midpoint of the tools
2. Compute the line that passes from the camera arm keyhole to the midpoint
3. Align camera to that line

```
function : track_midpoint
 keyhole_point = forward_kinematics(all ECM joint values
                                    set to zero)
 current_pose = forward_kinematics(current ECM joint
                                   values)
 current_direction = the vector from keyhole_point to
                     current_pose[position]
 psm1_point = forward_kinematics(PSM1 joint values)
 psm2_point = forward_kinematics(PSM2 joint values)
 mid_point = centroid of psm1_point and psm2_point
 desired_direction = unit vector from keyhole_point to
                     mid_point
 R = rotation matrix from current_direction to
     desired_direction
 L = extension length of the ECM's prismatic insertion
     joint
 new_ecm_point = keyhole_point + (desired_direction * L)
 ecm_pose[orientation] = R * current_pose[orientation]
 ecm_pose[position] = new_ecm_point
 new_ecm_joint_values = inverse_kinematics(ecm_pose)
```

AutoCamera Zoom Control Algorithm

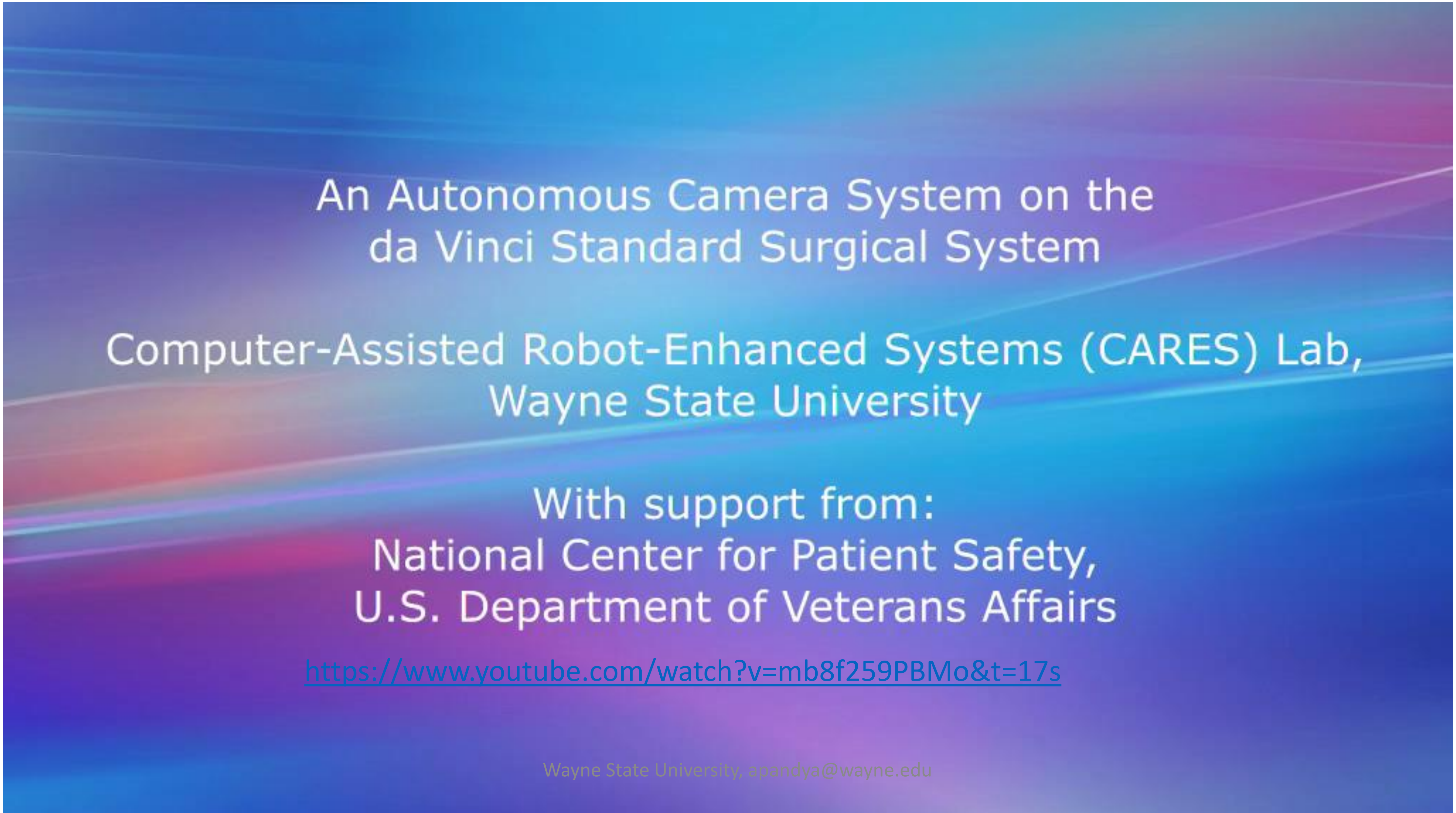
- Three zones around view center:



- The goal is to keep the tools in the dead zone
  - Zooms in when tool endpoints in inner zone
  - Zooms out when tool endpoints in outer zone

```
function : adjust_zoom_level
mid_point = x- and y-coordinates of the image’s center
dx = horizontal distance between either tool and the
     mid_point
dy = vertical distance between either tool and the
     mid_point
ax = horizontal distance between the mid_point and the
     edge of the image
ay = vertical distance between the mid_point and the edge
     of the image
if tool1 and tool2 are in the inner zone
    zoom_increment = min(dx/ax, dy/ay)
elseif tool1 and tool2 are in the dead zone
    zoom_increment = 0
elseif tool1 and tool2 are in the outer zone
    zoom_increment = -1 * min((ax-dx)/ax, (ay-dy)/ay)
zoom_level = zoom_level + zoom_increment
```

Implemented Baseline AutoCamera Algorithm

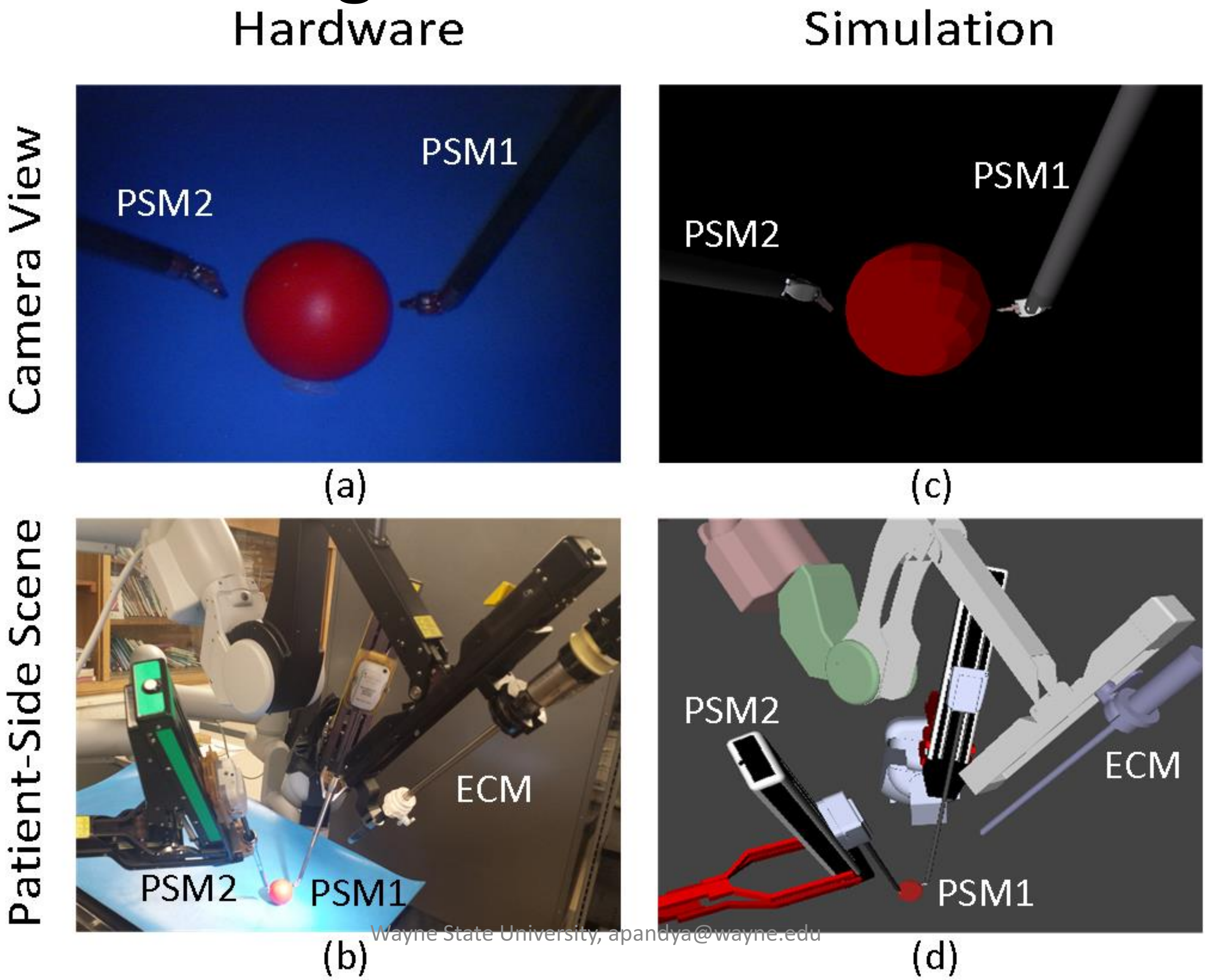


https://www.youtube.com/watch?v=mb8f259PBMo&t=17s

Basic Accuracy and Usability Testing

- System accuracy after co-registration was assessed by comparing the distance between tool tips for real hardware vs. ideal simulation in 10 poses
  - ECM to PSM1: mean absolute difference of 1.54 mm (95% C.I. of 0.40–2.68 mm)
  - ECM to PSM2: mean absolute difference of 1.91 mm (95% C.I. of 1.02–2.80 mm)
- We used the system to perform a peg-transfer task with AutoCamera
  - We were able to complete the task with much fewer interruptions than manual camera control
  - The basic AutoCamera algorithm followed the midpoints of the tools and zoomed as intended

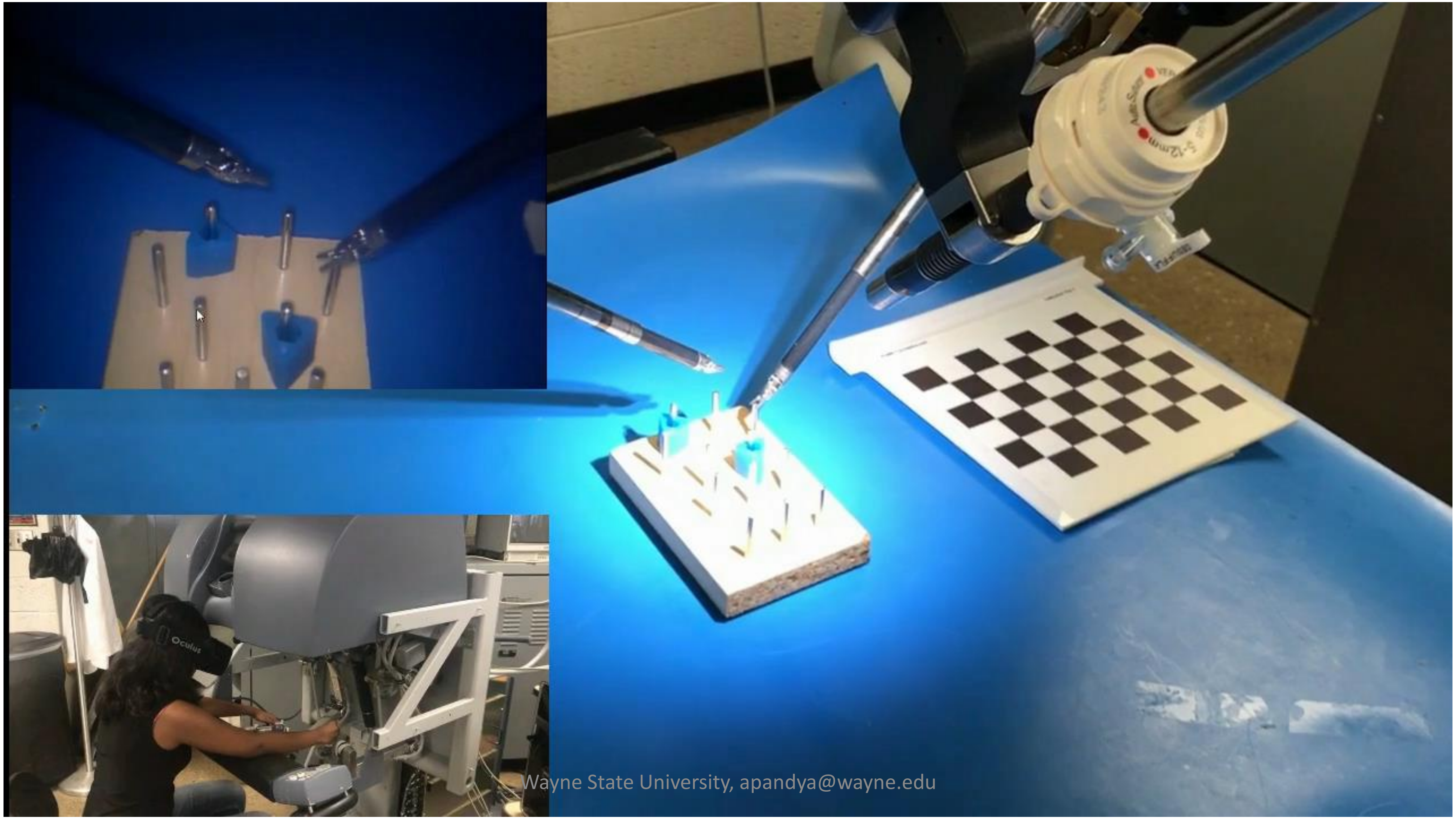
Results of Co-registration



da Vinci Recording and Playback System

https://www.youtube.com/watch?v=btgeu8B_qdQ

VR Headset for Camera Movement



Conclusions & Future Work

- AutoCamera on da Vinci is possible and has the potential to be useful
- The current AutoCamera algorithm is very simple
- More intelligent techniques, guided by testing, are needed
  - We're planning user subject studies to compare traditional camera control with AutoCamera
  - We're investigating the integration of task analysis and task-specific behaviors for different surgical procedures, including deep learning of tasks
  - We're considering imaging processing and other sensing techniques to support object tracking (bodily structures, clips, needles, etc.)

Credits & Collaborations

- Peter Kazanzides, Anton Deguet, Russ Taylor, and Zihan Chen, Johns Hopkins University
- Greg Fischer, Worcester Polytechnic Institute
- Simon DiMaio, Intuitive Surgical
- Michael Klein, M.D., Children’s Hospital of Michigan
- David Edelman, M.D., Detroit Medical Center
- Anthony Composto, M.S., WSU
- Tareq Dardona, B.S., WSU
- …and many other undergraduate/graduate students at the CARES lab

Feedback

Thanks for listening!

Any questions, comments, or suggestions?

Special Issue "Medical Robotics: Advances in Training, Ergonomics, Sensing, Control and Other Areas"

- A special issue of *Robotics* (ISSN 2218-6581).
- Deadline for manuscript submissions: **31 December 2017**
- http://www.mdpi.com/journal/robotics/special_issues/medicalrobotics
- A way to showcase dVRK and Raven research.
- In interested, email me apandya@wayne.edu