

Micron Range-Of-Motion Visualization

Student - Preetham Chalasani
Mentors - Dr. Russell Taylor, Marcin Balicki, Balazs Vagvolgyi
Department of Computer Science,
The Johns Hopkins University

INTRODUCTION

Micron is a fully handheld micromanipulator, which provides the surgeon a method to more precisely control μ -level hand movements. However, since the micron has a very limited range of motion, once it reaches the border, it starts to behave erratically. In this project I developed a visual alert assistance system for the surgeons dealing with very small anatomy, which will help the surgeons to keep a track of the micron motion, so that they don't let the micron hit the limits. To develop this application, I acquired variation information from micron and accordingly presented a visual feedback to the surgeons.



Fig. 1. Handheld micromanipulator, Micron

Application is mainly developed with the help of **CISST** & **SVL** libraries, for communicating with the micron and other components, like tool tracker, which helps me determine the pixel coordinates of the micron tip.

I assumed the workspace of the micron to be a cube or

a sphere, and displayed the tip position accordingly, on a 2D display screen.

BACKGROUND

Peeling delicate retinal membranes, which are often less than five microns thick, is one of the most challenging retinal surgeries. Preventing rips and tears caused by tremor and excessive force can decrease injury and reduce the need for follow up surgeries. So to address issues in the delicate micromanipulation of retinal membranes, a fully handheld micromanipulator, Micron was developed in the lab at Carnegie Mellon University, Pittsburgh.

Micron Setup

Unlike a robotic arm such as the SteadyHand, Micron is fully handheld micromanipulator and has motors

positioned between the handle grip and the instrument tip. The motors are three Thunder® piezoelectric actuators arranged in a star pattern, allowing for tip movement independent of hand motion within an approximate 1x1x1-mm range of motion centered on the handle.

Low-latency, high-bandwidth positioning information is obtained from custom optical tracking hardware named ASAP. Three LEDs on the actuated shaft of the instrument allow for 3 DOF (degree of freedom) position tracking of the Micron's tip. An additional LED attached to the handle provides hand movement information. Position sensitive detectors (PSDs) detect the LEDs within 1mm³ workspace.

The operator uses the micron under 18X microscope (Zeiss OPMI MD). A 27 gauge hypodermic needle with



Fig.2. System Setup

a 400 μ m shaft diameter is attached to the micron with a 5ml Nitinol wire inserted into the needle to serve as a very fine instrument tip. The Nitinol wire is tapered to a blunt point and painted white for enhanced contrast. Stereo cameras (PointGrey) attached to the microscope view the same workspace as the operator and capture video. The cameras track both the tip of the instrument and anatomical targets. The cameras are registered to ASAP so visual information regarding Micron's tip and tracked targets can be used for control. The entire setup is displayed below.

Utilizing stereo vision and tracking algorithms, the robot activates motion-scaled behavior as the tip nears the

surface, providing finer control during the critical step of engaging the membrane edge. It removes involuntary motion, such as tremor, by the actuation of the tip to counteract the effect of the undesired handle motion.

Significance and Importance

Microsurgical tools play a very significant role in very small and critical anatomies. Tremor compensation is a key component of many surgical robots and tries to eliminate the frequency bands dominated by tremor in order to remove unwanted motion while preserving the operators' intended movements.

Micron is one such microsurgical tool. It has been developed with actuators not on robot arm, but between the handle grip and the tool tip. Micron can move the tool tip independently of the hand motion, allowing it to perform behaviors such as tremor reduction, motion scaling and virtual fixtures.

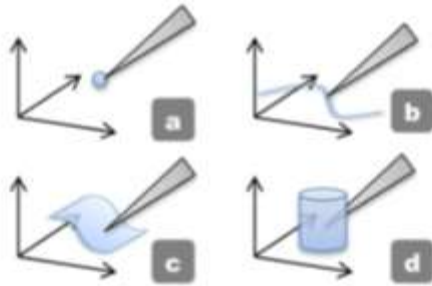


Fig.3. Virtual fixtures constraining the tip to a subspace with increasing degree of freedom: (a) Point, (b) Curve, (c) Surface, and (d) Volume.

However, the micron has a range of motion, in which the tool-tip moves. If the micron goes out of the range, then it starts moving in haphazard manner. And the surgeons would want to know before this happens.

GOAL OF THE PROJECT

For this reason, it's quite useful for the surgeons to have a visual assistance system, which would help them during small anatomies, where slight erratic movement of the micron might have dangerous repercussions.

There will be two main overlays:-

- Micron tip position in a 2D plane on which the tip is located.
- Depth of the micron tip from the home position.

Audio feedback is an experimental feature that I implemented.

Proposed Solution

Surgeons, while performing high profile surgeries, like vitreoretinal surgery, the slightest involuntary hand movement can turn out to be very dangerous. However, to negate those type of involuntary movements, just the use of micron won't suffice, surgeons have to know the tip position, along with the position inside the workspace, at all times.

Let's suppose, the micron tip is somewhere inside the retina at a certain depth, d . So, we draw a circle overlay that represents the 2D plane in which the micron tip is located, at depth d . Along with the circle overlay, a rectangular bar is drawn to indicate the height of the micron tip from its home position. A small red dot moves up and down indicating the depth of the micron tip from the home position.

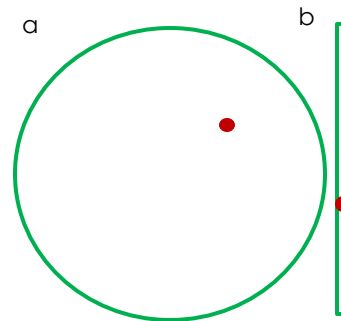


Fig. 4. Reddot represents the tip position and green outline represents the outer boundary of micron workspace. a) 2D plane display b) Depth display

TECHNICAL APPROACH

To develop the application, several technologies were used to successfully compile and integrate with the existing code. Following are few dependencies:-

- **CisstLibrary** - This library is the backbone of the project. All communications to and from the various components uses this library.
- **SVL** – cisstStereoVision is a part of Cisst libraries. It is optimized for real-time stream processing and visualization. Also features cross-network live video streaming.
- **MTS** - cisstMultiTask provides the component-based framework for the cisst package

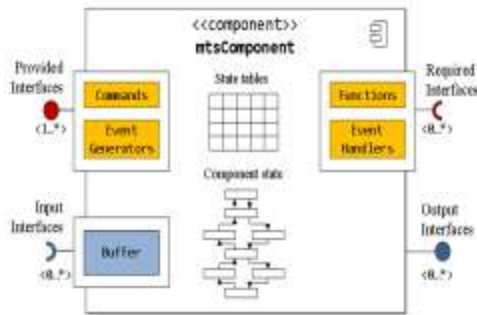


Fig. 5. Component Model

- **Scenario Manager** - This tool controls the interactions between various components. On startup, it finds all connected components and forms links between them.
- **ICE** - The Internet Communications Environment manages the connections between multiple computers. This simplifies the programming required to get computers to communicate with each other.

Procedure

A. Installation

To run the existing code base, a system with high configuration was required. Due to unavailability of the required configuration, a system with relatively low configuration was used to develop the application.

Step – 1 : Install cisst library along with all the supporting libraries like cisstMts, cisstSv etc.

Step – 2 : Install OpenCV-nonfree version. This library helps in drawing overlays on to the display screen.

Step – 3 : Install OpenAL and OpenGL libraries for audio and graphics, respectively.

Step – 4 : Install Fltk (Fast light toolkit) for creating various user interfaces.

B. Circle Overlay

The first task of the project was to create a sample circle overlay and move a red dot inside the circle. These were created using `svlStaticEllipse`. The big circle was created with a radius of 100px (for testing) and the small red dot with a radius of 5px. Then to display these overlays on the screen, one has to add these overlays to the overlay filter, which takes care of pushing the overlays on to the screen. Also, the overlay display task was added to the provided interface of the `micronLimitsBehavior`, which is a part of

`visualizationprocess` component. After the successful display of circle and red dot on the screen, random data was generated to mimic the position of the microntip.

C. Painter Test

Painter is an existing feature of the application. This behavior is also an interface of `visualizationprocess` component. It helps the user to draw on the screen with the help of the mouse pointer. This feature was used to mimic the microntip position on the screen.

Firstly, `micronBehaviorLimits` and painter were connected, so that the painter interface can use the methods of `micronBehaviorLimits` to update the overlays, whenever user tries to draw something on the screen.

D. Alert System

Now, once the `micronBehaviorLimits` receives various positions of the simulated microntip, it scales down to make the red dot move in the circle overlay, that was created at first.

Assuming the range of the micron to be 100px and the radius of the circle also to be the same as the range, we have a red dot, acting as the micron tip, moving inside the workspace, with the home position as the center of the circle. Now, as the red dot moves closer to the boundary of the circle, the thickness and the color of the boundary changes, helping the user to maintain the micron tip in its range of motion.

E. Circle Overlay for Home Position

Now, another circle overlay is drawn at the center of the screen (for testing purposes), representing the home tip. This overlay, unlike the previous one, moves with respect to the home location, making the tip location constant at the center of the screen. This overlay is also tested with simulated micron data from the painter.

F. Random Data from devMicron

DevMicron is the component which receives and sends information of the micron to other components. This component has only one dependency, i.e, it has to run in the same network as the system to which the micron is connected. Its provided interface has various methods, out of which following are few required methods for `micronBehaviorLimits` :-

- `getHomeTipInAsap`
- `getTipInAsap`
- `getHomeTipInMicronFrame`
- `getTipInMicronFrame`
- `getTipInHomeFrame`

The component has the position of the micron tip and home location in 3 different coordinate system.

- **AsapFrame** :- Asap is a custom optical tracking hardware which provides low-latency, high-bandwidth positioning information.
- **MicronFrame** :- This frame located at the handle to which the needle is attached.
- **HomeFrame** :- This frame is located at the home position of the micron. Home position can be assumed as the micron tip location when the cancellation is off, i.e, micron is off.

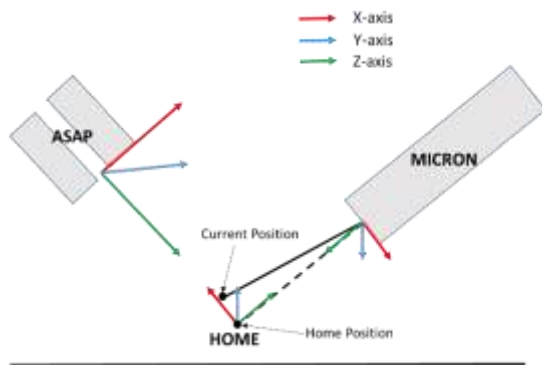


Fig. 6. Coordinate Frames

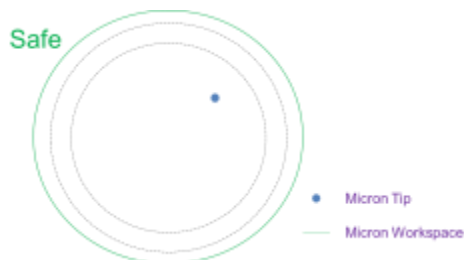
Next, random tip location for ASAP frame was generated and passed to the `micronBehaviorLimits` for component connection and also overlay display.

G. Audio Feedback

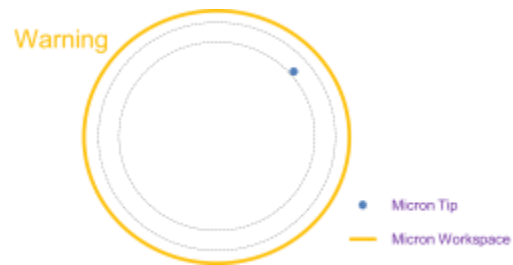
Once, the alert system is integrated, implementing audio functions is quite easy. Keeping in mind the delay of the tool tracker, 3 different regions, inside the workspace, were selected for efficient feedback.

Let's assume the tip position from the center of the circle is T pixels and the radius of the circle overlay is R , then the three regions can be categorized as

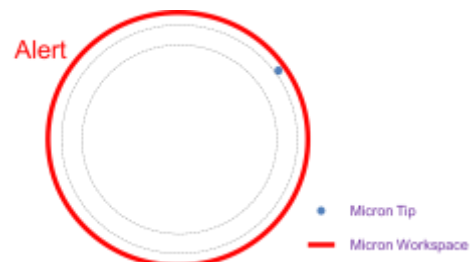
- **Safe** :- $T < (R - 15)$



- **Warning** :- $(R - 15) < T < (R - 8)$



- **Alert** :- $T > (R - 8)$



When the micron tip is in the safe zone, there would be no warning sound. However, when the tip crosses the safe zone and enters the warning zone or, later, alert zone, two different audio clips will be played for warning and alert, respectively. Even if the tip crosses the circle, it is assumed to be in alert zone.

H. Cube workspace of the micron

Sphere workspace was assumed previously for the ease of coding. However, the workspace of the micron is more likely a cylinder or a cube. Thus, cube workspace has also been developed, so that in future, application can be switched to use cube or a sphere or a cylinder. Since cube is a 3 dimensional object and the display is 2 dimensional, isometric projection was used to display 3D cube on a 2D screen.

Isometric Projection:- It is a method for visually representing 3-dimensional objects in 2-dimension. It is an axonometric projection in which the three coordinate axes appear equally foreshortened and the angles between any two of them are 120 degrees.

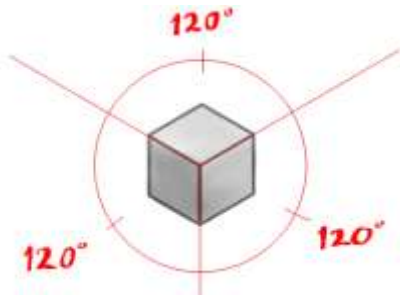


Fig. 7. Isometric projection of a cube

So, a cube overlay was drawn in 3-dimension, around the home tip and projected on the 2-dimensional display screen.

I. ASAP Orientation

The ASAP cameras are not always fixed at a particular position. The overlays have to change the orientation according to the ASAP orientation. We consider the three critical flight dynamics parameters known as *roll*, *pitch* and *yaw*.

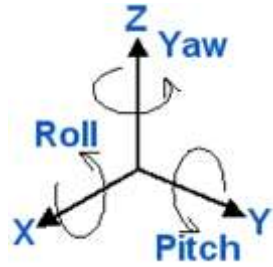


Fig. 8. Angles of Rotation

- Yaw – A *yaw* is a counterclockwise rotation of α about the Z-axis. The rotation matrix is given by

$$R_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- Pitch – A *pitch* is a counterclockwise rotation of β about the Y-axis. The rotation matrix is given by

$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}.$$

- Roll – A *roll* is a counterclockwise rotation of γ about the X-axis. The rotation matrix is given by

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}.$$

J. ASAP Gui

The ASAP gui was created using the fltk libraries. This interface helps the user to change the scale values according to their interest. Also the orientation of the ASAP can be managed using this interface. It looks like the figure displayed below :-

K. Depth Display

To display the depth or height of the micron tip from its home position, a rectangle overlay is been used. This

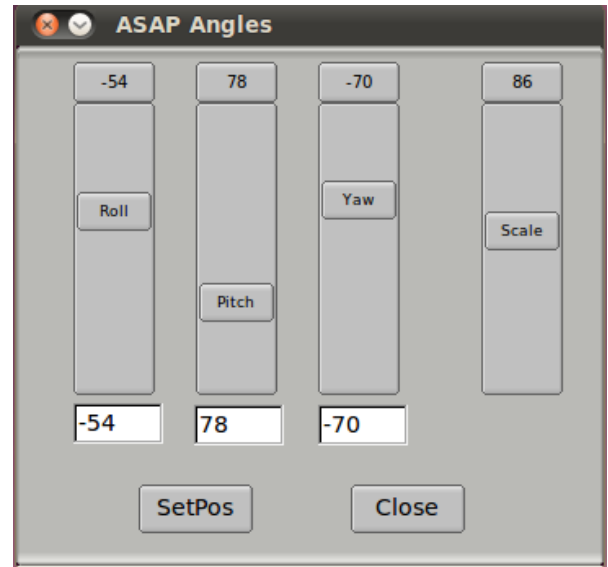


Fig. 9. ASAP Interface

overlay is fixed on the screen with towards the right of the micron-tip-circle-display. However, the size of the overlay will change with the change of scaling value.

L. Scaling

Finally, scaling of the overlays. We don't want our overlays to have fixed dimensions. Scaling of the overlays means changing the value of pixels per mm. Dimensions and size of all the overlays will change with the change of scale value.

EXPERIMENTAL DATA

Few experiments were conducted with two types of microns.

- 6DOF – Cylindrical workspace
- 3DOF – Cubical workspace

For the 6DOF micron, the overlays created were sufficient enough. This experiment was conducted, without the audio feedback and the depth/height display.

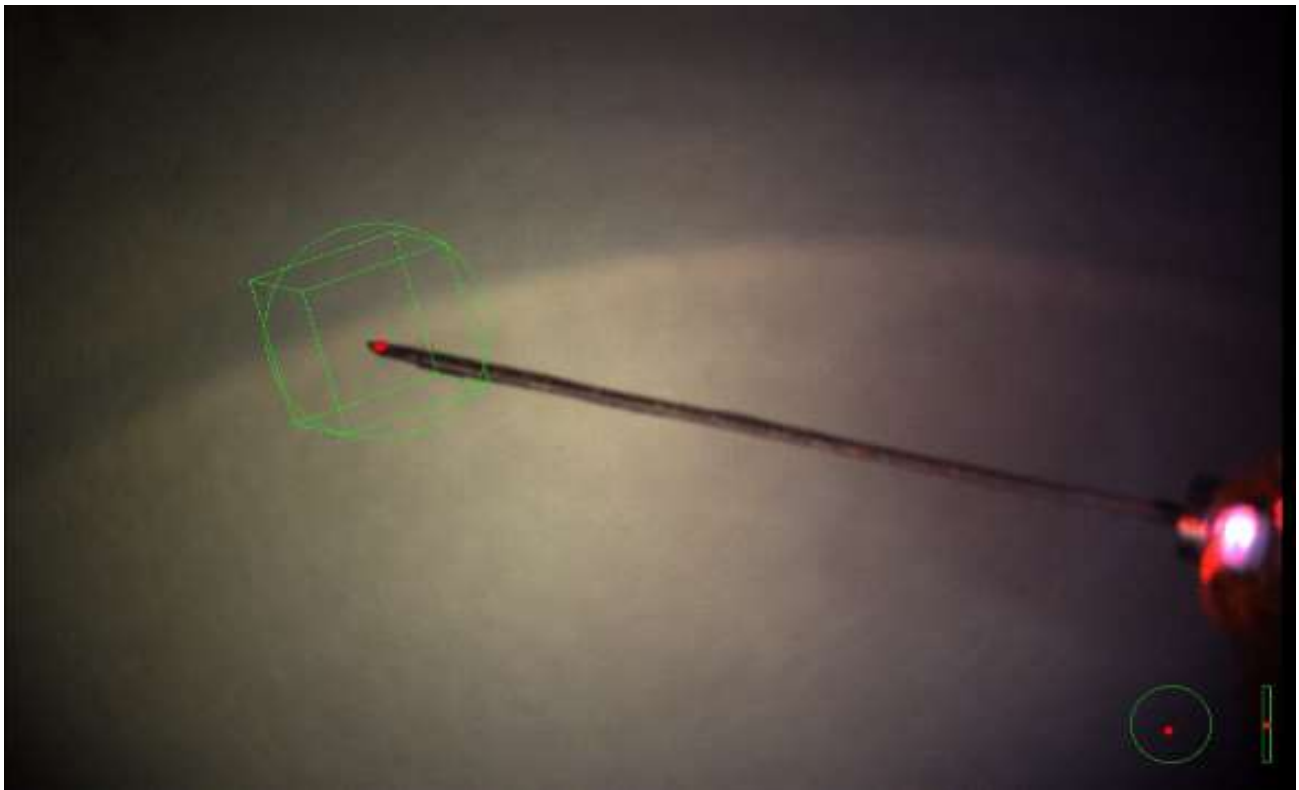


Fig. 10. Screenshot 1 - Safe Zone

However, the micron-tip-circle-display was working well. Following are the dimensions of the workspace and the orientation of the ASAP camera at which the micron was tested:-

- Workspace
 - Height : 4mm
 - Radius : 2mm
- Scale : 1mm = 100px
- ASAP Orientation
 - Yaw : 22.5
 - Pitch : 50
 - Roll : -10

For the 3DOFmicron, all the overlays were tested and were running perfectly. The experiment was conducted with the audio feedback and also with the depth and cube display. Following are the dimensions of the workspace and the orientation of the ASAP camera at which the micron was tested:-

- Workspace – $1mm^3$
- Scale – Different scale values
- Orientation – Various orientations

Fig. 10, 11 and 12 are few screen shots of the application.

DELIVERABLES

Following are the deliverables that were discussed in the beginning of the project, which have been successfully implemented and tested:-

Minimum deliverables

- Develop a test application with some overlays.
- Test the application with random simulated data.
- Communicate with the micron and retrieve the information
- Display the information according to the overlays

Expected deliverables

- Develop the visual assistance system
- Develop an overlay to display depth of the micron tip
- Integrate the tool tracker into the application
- Get feedback from the surgeons

Maximum Deliverables

- Audio Feedback
- Develop an overlay to display the workspace of the micron
- Conduct rigorous testing

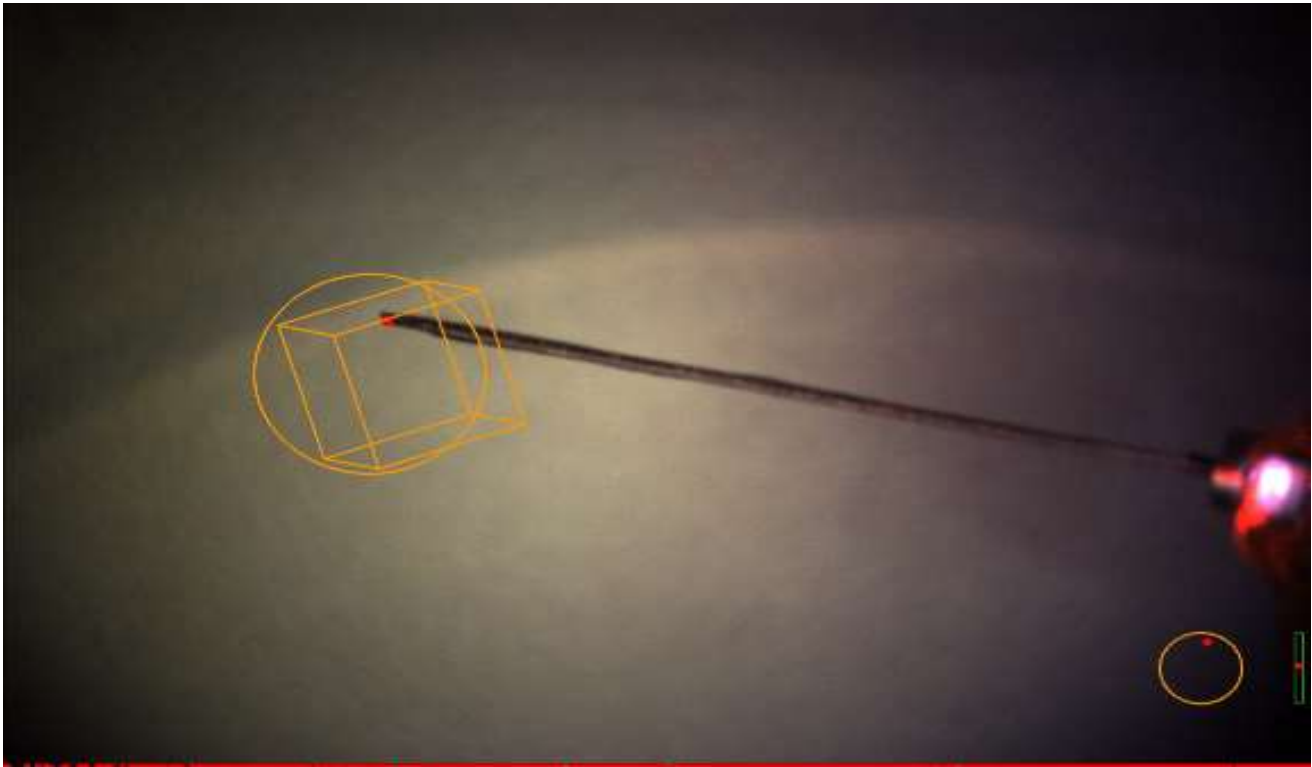


Fig. 11. Screenshot 2 – Warning Zone

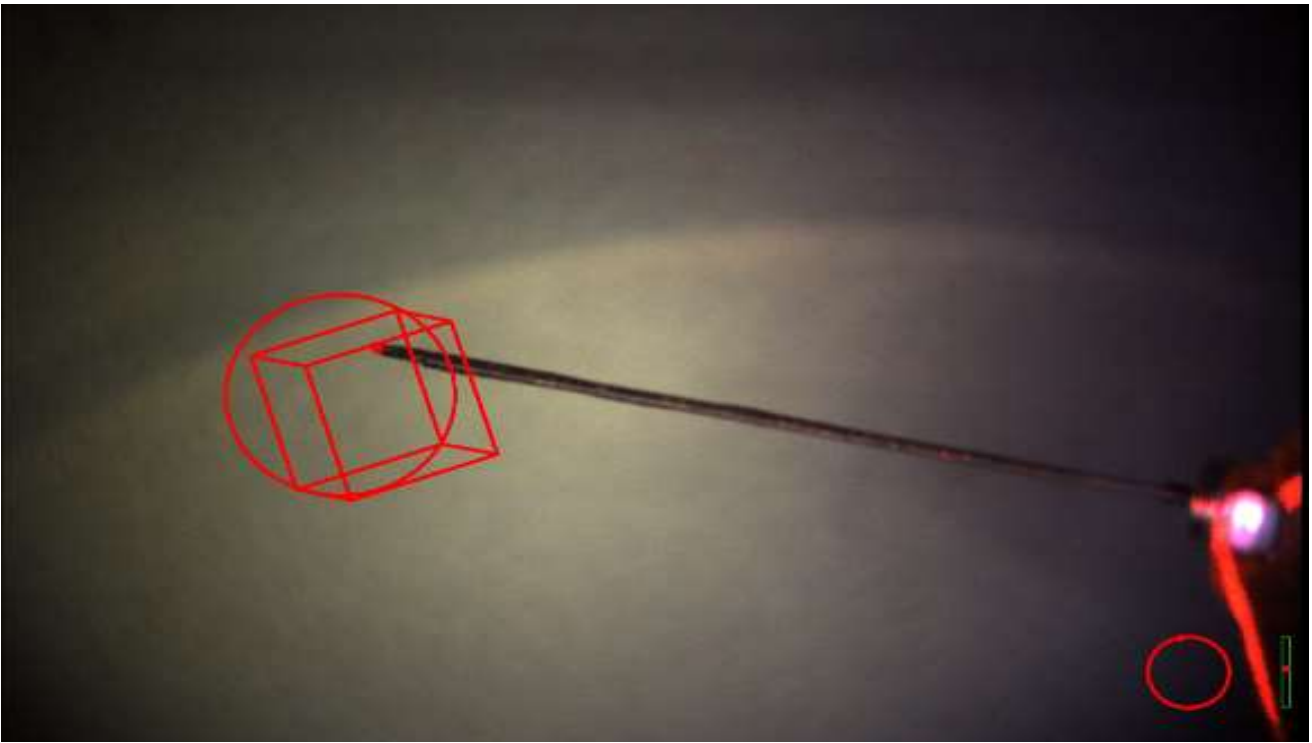


Fig. 12. Screenshot 3 – Alert Zone

RESULTS

On rigorous testing and debugging, I feel that this application is ready to be integrated in the existing code base and use of this application will be very helpful for research purposes. Furthermore, the application was tested during a bunny experiment. The picture shown below display the experimental setup and this experiment gave me some ideas for future work, and overall it was a success.



FUTURE WORK

- ASAP orientation can be retrieved, initially, from the gyro sensors. Later, the same ASAP interface can be used to control the ASAP.
- Workspace of the micron is a 2D isometric projection, which can be rendered as a 3D figure.

PROBLEMS FACED

- Access to micron – Not always did I have the access to the micron.
- Build/ Compile delay – There are a lot of sleeps in the code, which delays the build time, as well as the compile time.
- 3D rendering of cube – Initially, I tried to render a 3D cube instead of a isometric projection of a 3D cube, which was a bit problematic, since cisst doesn't support 3D rendering. Also, trying to integrate OSG(open scene graph) at the last moment was a bit risky.
- Integration of the tool-tracker

MANAGEMENT SUMMARY

I spent on an average of 16 hours per week on this project. I had weekly meetings with my mentors and progress reports discussed in the meetings. Following are few important milestones and their completion dates:-

- ✓ Setting up the development environment and understanding the existing framework.
 - Feb 18, 2013

- ✓ Test application with working overlays
 - March 11, 2013
- ✓ Have the basic overlay communicating with micron for the bunny experiment.
 - March 29, 2013
- ✓ Implement a way to display the depth of the micron tip and the workspace of the micron
 - April 29, 2013
- ✓ Integrate the tool tracker and rigorous testing
 - May 4, 2014

ACKNOWLEDGEMENTS

I would like to thank my mentors for their support throughout the project. Their input and guidance was immeasurable. I would specially like to thank Balazs for his valuable help and suggestions along the way.

REFERENCES

- [1] B. C. Becker, S. Voros, R. A. MacLachlan, G. D. Hager, and C. N. Riviere, "Active Guidance of a Handheld Micromanipulator using Visual Servoing", in IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12-17, 2009. pp. 339-344.
- [2] B. Becker, R. MacLachlan, and C. Riviere, "State estimation and feedforward tremor suppression for a handheld micromanipulator with a Kalman filter", in IEEE RSJ Int Conf Intell Robot Syst, 2011. pp. 5160-5165. NIHMSID: 345014.
- [3] B. Becker, R. MacLachlan, L. Lobes, and C. Riviere, "Vision-Based Retinal Membrane Peeling with a Handheld Robot", in IEEE Int Conf Robot Autom, 2012. pp. 1075-1080. NIHMSID: 368417.
- [4] B. Becker, S. Yang, R. MacLachlan, and C. Riviere, "Towards vision-based control of a handheld micromanipulator for retinal cannulation in an eyeball phantom", in Proc IEEE RAS EMBS Int Conf Biomed Robot Biomechatron, 2012. p. accepted for publication. NIHMSID: 368431.
- [5] B. Gonenc, M. A. Balicki, J. Handa, P. Gehlbach, C. N. Riviere, R. H. Taylor, and I. Iordachita, "Preliminary Evaluation of a Micro-Force Sensing Handheld Robot for Vitreoretinal Surgery", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal, 7-12 October, 2012. pp. 4125-4130.
- [6] R. MacLachlan, B. Becker, J. Cuevas-Tabarés, G. Podnar, L. Lobes, and C. Riviere, "Micron: an actively stabilized handheld tool for microsurgery", IEEE Trans Robot., vol. 28- 1, pp. 195-212, 2012. NIHMSID:345015.
- [7] S. Yang, M. Balicki, R. A. MacLachlan, X. Liu, J. U. Kang, R. H. Taylor, and C. N. Riviere, "Optical Coherence Tomography Scanning with a Handheld Vitreoretinal Micromanipulator", in IEEE Engineering in Medicine and

Biology Conf, San Diego, Aug 28-Sep 1, 2012. pp. 948-951. NIHMSID: 383510.

- [8] S. Yang, R. MacLachlan, and C. Riviere, "Design and analysis of 6 DOF handheld micromanipulator", in Proc IEEE Int Conf Robot Autom., St. Paul, MN, May 14-18, 2012. pp. 1946-51. NIHMSID: 368427.
- [9] B. Becker, R. MacLachlan, L. Lobes, G. Hager, and C. Riviere, "Vision-Based Control of a Handheld Surgical Micromanipulator with Virtual Fixtures", IEEE Transactions on Robotics, pp. Accepted Nov 27, 2012, 2013. NIHMSID: 429749.
- [10] M. Balicki, J.-H. Han, I. Iordachita, P. Gehlbach, J. Handa, R. H. Taylor, and J. Kang, "Single Fiber Optical Coherence Tomography Microsurgical Instruments for Computer and Robot-Assisted Retinal Surgery", in Medical Image Computing and Computer Assisted Surgery (MICCAI 2009), London, September 20-24, 2009. pp. 108-115. PMID: 20425977