HMD-Based Navigation for Ventriculostomy

Mingyi Zheng, Yiwei Jiang, under the auspices of Professor Peter Kazanzides and Ehsan Azimi

***Abstract*— We provided an HMD-based navigation system to introduce image guidance via augmented reality on Microsoft HoloLens to improve the success rate of catheter placement in ventriculostomy. The proposed system includes a ZED mini camera mounted on HoloLens to provide a larger field of view and software for AR overlay of a ventricle model as well as a catheter tracking algorithm. Pilot test results shows significantly higher scores with our system, which demonstrates the feasibility of our system.**

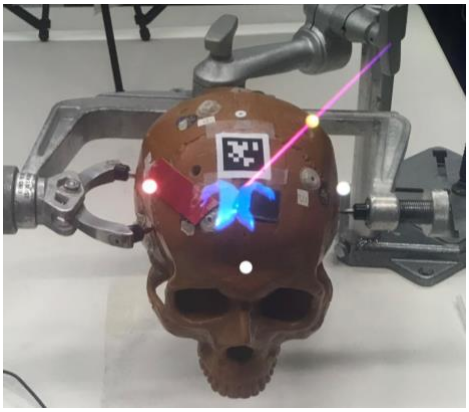


Fig. 1 User view

I. Introduction

Ventriculostomy is a common procedure in neurosurgery. The process of the surgery requires creating a hole within a cerebral ventricle and inserting a catheter to drain excess fluid. In [4], the authors state that there is considerable variation among neurosurgeons with regard to entry points. In [5], the authors indicate that despite the frequency and standardization of this procedure, suboptimal placement or misplacement of the catheter after the first passage and the final passage occurs in up to 23%–60% of cases, when using the freehand technique on the basis of anatomical landmarks. Moreover, the location of the ventricle sometimes shifts, which makes the surgery more challenging.

II. METHODS

A. *System Overview*

Our system consists of a PC (yellow), HoloLens (black), catheter (blue) and skull phantom (orange) shown in the Fig. 2 below. Firstly, our system is assuming that the ventricle is static. And we have a tool with AR marker on the end to localize anatomical points for ventricle and skull registration. There is a pivot calibration [1] for accurate tool tip tracking. The display calibration [2] is also implemented to ensure the accuracy of AR overlay on HoloLens with an external camera. PC is responsible for ventricle and skull 3D-reconstruction (green) as well as marker and catheter tracking system in ZED mini camera (gray). The coordinates from the tracking system are sent to Unity through UDP in real-time while the segmentation and 3D reconstruction are implemented offline in 3D Slicer based on CT/MR image. Once HoloLens receives the data, it starts generating an AR overlay to display the image guide on the skull phantom.

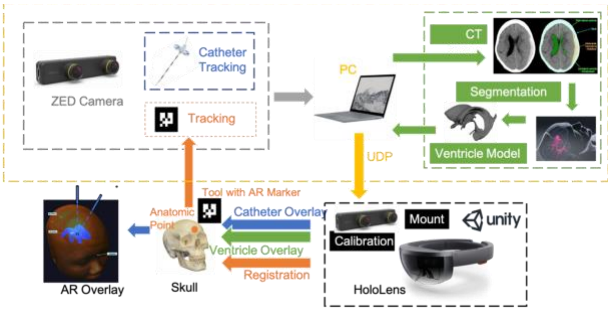


Fig. 2 System overview

B. *Workflow*

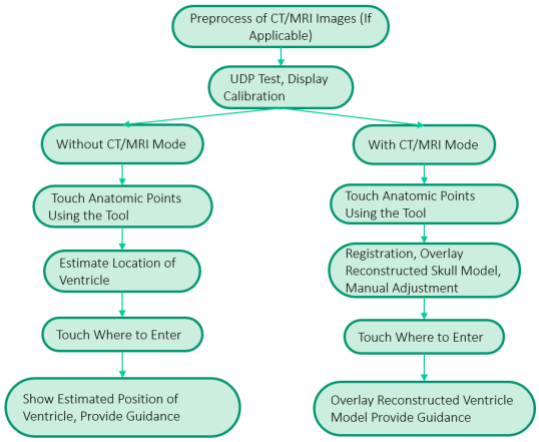


Fig. 3 Workflow

C. *Hardware*

We designed a camera mount on HoloLens, shown in Fig. 4, so that the Zed Mini camera is integrated for catheter tracking and marker tracking while providing a larger field of view and higher resolution and accuracy. The base is designed

to have the same curvature profile as the HoloLens top surface for attachment via double-side tape. The revolute joint design allows the camera to be adjusted in the pitch axis. The yaw axis is also adjustable by screws but requires re-assembly.

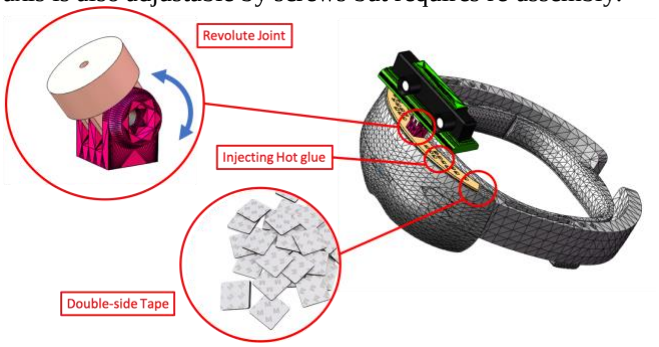


Fig. 4 Camera mount for Zed Mini

D. *Software Architecture*

The software of our system mainly consists of two parts, one is on the PC, where we use the ZED SDK and ArUco in OpenCV with Python for AR markers tracking and depth sensing, and another part is a Unity application running on HoloLens, regarding registration, AR overlay and UI. Communication between PC and HoloLens are through UDP protocol.

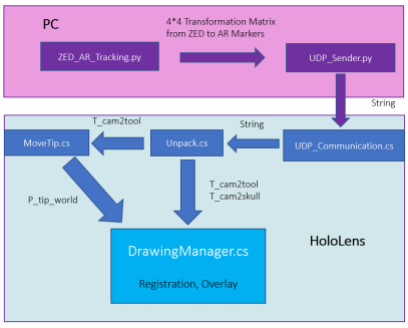


Fig. 5 Software architecture

E. *Calibration*

Two methods have been utilized to calibrate two parts in our systems. We used Pivot Calibration [1] for obtaining the location of the pointer's tip relative to the marker which is tracked. Display Calibration [2] is for OST-HMD overlay while using ZED Mini as tracker. This calibration properly aligns the coordinate system of a 3D virtual scene that the user sees with that of the tracker, in our case, ZED Mini.

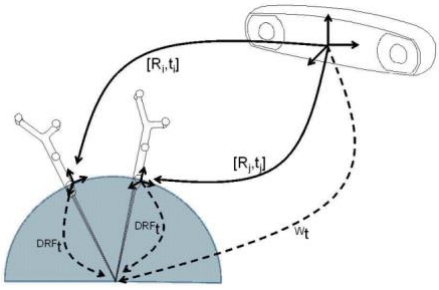
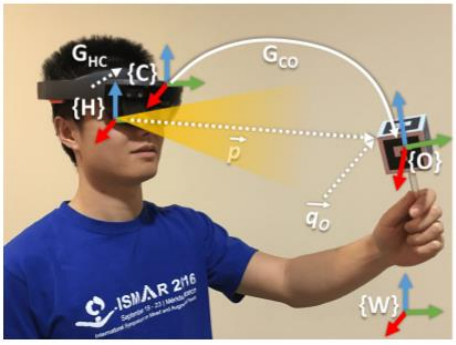


Fig. 6 (a) Schematic of pivot calibration



(b) Display calibration

F. *Registration*

Based on the existing clinical workflow, we designed this Three-Point Registration process to register the virtual skull model to the real skull phantom. The user uses a pointer touch the glabella and temples on both sides of the skull; meanwhile, the coordinates of these three points are obtained in the Unity world frame, then we build the anatomical coordinate system in Unity so we can register the virtual skull model to the real skull phantom. However, three points is the minimum requirement for registering two 3D objects, the anatomic points on the human skull are not specifically defined, and the result of this process is prone to error, so we added a Manual Registration process as an adjustment, allowing users to translate and rotate the virtual model, aligning the overlay to its real counterpart like a surface to surface registration.

At first, we have another marker fixed on the skull phatom, when user is pointing these three anatomic points, we read the coordinates of them in this skull marker's frame rather than Unity world frame, but we suffered a lot from the jitting due to the noise from AR marker tracking, so we changed to use the Unity world frame as our reference. But this method also has its limitation because it relies on SLAM of HoloLens, which is not very accurate. We may try to use Kalman Filtering and combine these two methods for better performance in the future.

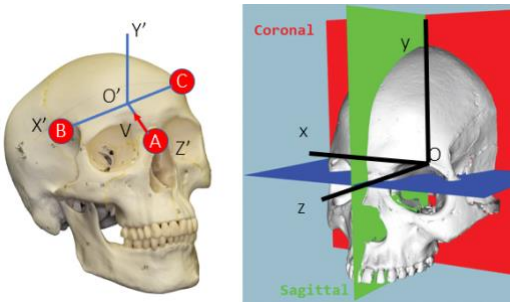


Fig. 7 Anatomical coordinate system

Fig.8 (a) After Three-Points Registration

(b) After Manual Registration

G. *Ventricle & Skull Segmentation and reconstruction*

Ventricle and skull segmentation and 3D-reconstruction are implemented in 3D Slicer as shown in Fig. 9 Segmentation and 3D-reconstruction. The procedure is the following:

1. Choose a threshold for ventricle/skull
2. Select target object
3. Close holes
4. Smooth and mesh
5. Generate 3D model

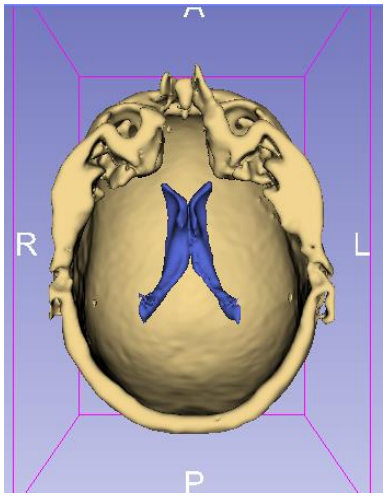


Fig. 9 Segmentation and 3D-reconstruction

H. *Catheter Tracking*

The catheter tracking algorithm is modified based on the surgical tool tracking algorithm in [3]. The algorithm assumes neurosurgeons always wear colored gloves, the entry point is known, and the catheter is straight and rigid. The algorithm is designed as follows:

1. Hand tracking: convert image to HSV to extract the purple glove color and pick the two largest areas as candidates
2. Mask: confirm the hand tracked in the 1st step is correct by checking if the candidate has similar depth within its region. If both candidates have similar depth, then pick the one with largest area. If not, pick the one with similar depth. Make a mask of the region around the hand as shown in the left of Fig. 10
3. Find catheter edge line: Probabilistic Hough Transformation (PHT) implemented and filter lines by length and minimum gap between found points.
4. Find catheter endpoint: the endpoint is calculated by averaging the filtered endpoints in the 3rd step above
5. Catheter angle and insertion depth: After getting 2D coordinates of the endpoint, the depth can be extracted from the Zed mini camera. Then, the angle error and insertion depth can simply be calculated by the following formula

$$\vec{p} = endpoint - entrypoint$$

$$depth = ||\vec{p}||$$

$$\theta_{\text{angleerror}} = \cos^{-1}\frac{\vec{p}\cdot\vec{g}}{||\vec{p}||\,||\vec{g}||}$$

where $\vec{g}$ is guide linie vector from ventricle center to entrypoint

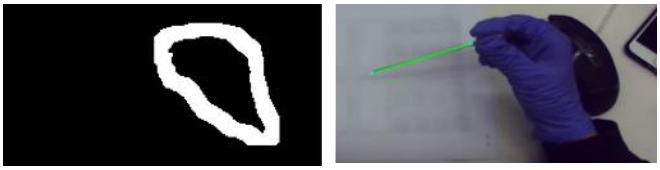


Fig. 10 Catheter Tracking, the blue dot is tracked endpoints and the green line is the line detected by PHT

I. *User Study Protocol*

A test board with 8 alternative positions and an adaptor mount to the skull for hex standoffs is shown in Fig. 12. The hex standoff design is to ensure the repeatability of the experiment so that the ventricle phantom orientation is fixed when the ventricle phantom position is changed.

Fig. 11 is a sectional view of the skull phantom; we had magnets press fit into the skull phantom and tape foam on the holes for catheter entry to make the feeling of insertion be as close as possible to the real brain.

The experiment is set up by choosing one of the 8 alternative locations for the ventricle; in our test case, we used the same position shown in Fig. 12 for both users. As shown in Fig. 13, the user wore HoloLens to perform catheter insertion on the skull phantom 5 times and we recorded the number of times that the ventricle was hit to evaluate the success rate. The point score is evaluated by the following criteria based on Fig. 11 score evaluation:

- Red region: 20
- White region: 10
- Other: 0

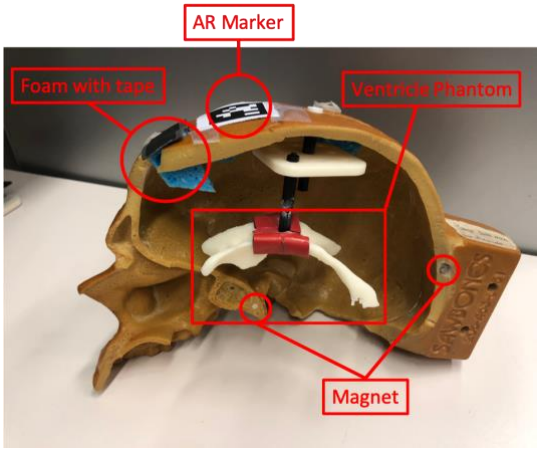


Fig. 11 Sectional view of the skull phantom

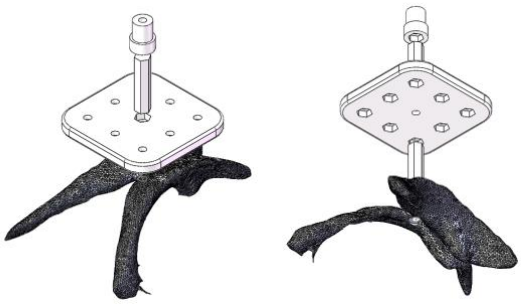


Fig. 12 CAD model of mount for user study

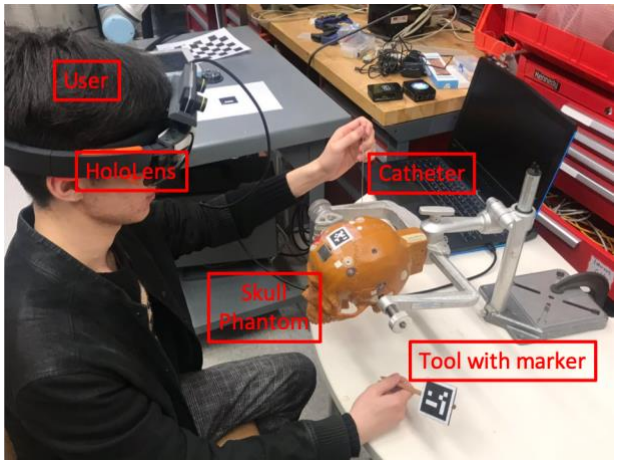


Fig. 13 Pilot Test

III. ACCURACY TESTS AND EVALUATION

Overall accuracy of the whole system consists of several parts. We tested and evaluated four specific components as follows.

A. AR Marker Tracking

The ArUco package in OpenCV with Python is used in our system for AR marker tracking. For the test, we glued a marker to one surface of a 2-inch cube, printed a sheet with 5*3 grids with side length of 5 cm, and align the corner of the cube with the corner of the grid on the paper. First we did the test in the X axis, which is perpendicular to the marker plane; after one alignment we move to the next adjacent corner in the X axis, repeated 12 times (Case 1). Then we did the same procedures in the Y axis which is parallel to marker plane (Case 2). We also tried if we change the lighting condition, shut down the flash light, the error becomes larger. Due to the time limit, we haven't gotten chance to thoroughly investigate how the lighting affects the AR tracking error. We plan to study this by setting up experiements in the future work.

The results show that, when dealing with depth, the standard deviation is larger, but the mean is better (Fig. 15 Case 1). From the plot we can see that there must be a systemic error in tracking displacement in the y axis because all the data points are below 5cm (Case 2). We also tried if we change the lighting condition, shut down the flash light, the accuracy is worse(Case 3).

Fig. 14 Marker tracking test setup

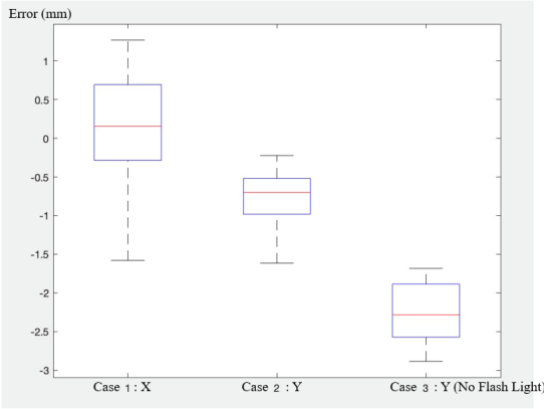


Fig. 15 Marker tracking error

Error	Axis X(mm)	Axis Y(mm)	Axis Y(mm) No Flash Light
Mean	0.120	-0.766	-0.2238
Std	0.0801	0.0345	0.0414

TABLE 1 Marker tracking results

B. Pivot Calibration

We kept the tip of the pointer at a fixed location on the table, rotate the pointer around this location to five poses, for

each pose we read the transformation matrix from camera to marker. Then we computed the tip coordinate in marker coordinate system, optimal in least square sense.

$$\begin{pmatrix} [R1][-I] \\ [R2][-I] \\ \cdots \\ \cdots \\ [Rn][-I] \end{pmatrix} \left(\begin{bmatrix} \vec{p}_{\text{tip}_{\text{marker}}} \\ \vec{p}\_tip\_cam \end{bmatrix} \right) = \begin{pmatrix} -t1 \\ -t2 \\ \cdots \\ -tn \end{pmatrix}$$

$$\vec{p}\_\text{tip\_marker} = \begin{bmatrix} 12.33948137 \\ 0.09685484 \\ -0.79652977 \end{bmatrix} \text{(cm)}$$

$$\vec{p}\_tip\_cam = \begin{bmatrix} -5.27035525 \\ 9.09906792 \\ 42.44843996 \end{bmatrix} \text{(cm)}$$

$$\Sigma_{residual} = 0.51383263 \text{ (cm)}$$

C. AR Display Calibration

According to [2], the mean of reprojection error of display calibration, using isometric model, is 5.86mm and the standard deviation is 0.81mm. We achieved the same level of accuracy by eye inspection.

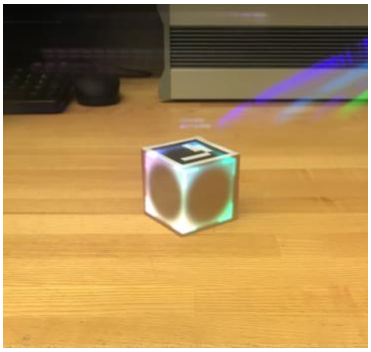


Fig. 16 Display calibration result

D. Tool Tip Tracking

This test is about tool tip tracking accuracy. We attached a marker to the same paper as in Test A, then use the pointer to touch 20 points on the grid, then compute the distance from the marker's center to the tip of the pointer; for each point we took 10 measurements. The true distances are given by design and verified with a ruler.

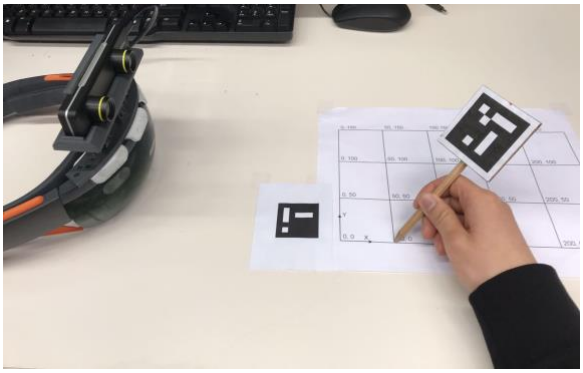


Fig. 17 Tool tip tracking test

Error(mm)		x-axis				
		50mm	100mm	150mm	200mm	250mm
y-axis	50mm	-3.47	-3.12	-5.75	-3.51	-7.39
	100mm	2.91	2.55	1.17	1.07	-1.09
	150mm	-2.26	-0.59	4.49	5.91	5.26
	200mm	4.05	9.26	12.41	19.13	21.37
Error std(mm)		x-axis				
		50mm	100mm	150mm	200mm	250mm
y-axis	50mm	2.82	2.05	2.59	1.88	2.13
	100mm	1.63	1.63	5.09	0.80	0.91
	150mm	1.36	1.38	1.09	1.39	1.15
	200mm	0.91	0.92	1.12	0.82	1.08

TABLE 2 Distance error results
(From the marker's center to the tip of the pointer)

The average measurement error is $3.12\text{mm} \pm 8.52\text{mm}$. From the results we can see that the error at 200mm in the y-axis is significantly larger than the others. We believe that this is because that the marker at 200mm in the y-axis is at the edge of the camera field of view so that it has larger distortion, we think that it may be because the calibration of ZED Mini is not very good.

E. Virtual Tool Tip Displacement in Unity

Similar to Test A and Test D, we use the pointer to point to one of the corners on the paper, read its coordinates in the Unity world frame, and then move the tip to another neighbor corner, then compute the distance. Repeat this procedure for 15 times.

We did another group of tests in which HoloLens was worn by a user rather than placed on the table, after each move, the user also changed position.

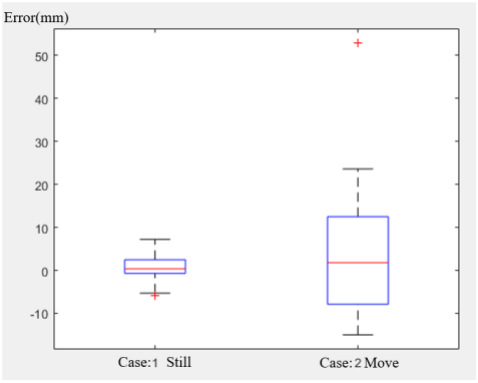


Fig. 18 Displacement error in Unity

	Fixed (mm)	Move (mm)
Error Mean	0. 8	4.5
Std	3.5	17.7

Table 3 Error Stats

Comparing the two datasets, we can conclude that if the HoloLens does not move, we are able to track the tip of the pointer quite well, but once it moves, the error increases obviously.

The ultimate accuracy of our system is how coincident we overlay the virtual ventricle model to its real counterpart, however, it is challenging for us to quantify this result since we have not found a way to measure the error objectively, also

only the user behind HoloLens can see the results. Furthermore, due to drift problem, the error changes as the users moves, so it is still difficult to evaluate.

IV. PILOT TESTS AND RESULTS

From the Table 4 below, it shows both users get significant improvement in terms of the total score. Without our navigation system, both users hardly ever hit the red(center) part of the ventricle phantom. However, with the navigation system, both users hit the red(center) part of the ventricle phantom once they hit the target once. However, the data is only collected with 2 users with fixed ventricle position. More formal tests in the future should have more users while changing the position of the ventricle phantom after each trial.

Without HMD-based Navigation System						Evaluation
Trial #	1	2	3	4	5	Total Score (100)
User 1	0	20	10	0	10	40
User 2	0	0	10	10	10	30
With HMD-based Navigation System						Evaluation
Trial #	1	2	3	4	5	Total Score (100)
User 1	0	20	20	20	20	80
User 2	20	20	20	20	20	100

Table 4 Pilot Test Results

V. CONCLUSIONS AND FUTURE WORK

From what have been discussed and the results above, although our test setup is still very different from surgical conditions, it is safe to conclude that this navigation system helps users to perform this mock ventriculostomy.

However, there are a few limitations of the system. First, we experienced AR overlay drift issue, it may be related to the display calibration. And depth from the Zed mini camera is inaccurate for the reflective material of the catheter coating. Additional, AR marker tracking is influenced by lighting.

Limited by immaturity of technology and performance of hardware, our system still has many defects, but we believe that our project has a future that ultimately can help surgeons improve their success rate of this procedure in real surgeries.

Yiwei will continue working on this project during this summer. Machine Learning methods will be utilized for segmentation of skull and ventricle. Depth information probably will be taken advantage of for registration based on ICP. The catheter tracking algorithm needs to be improved, tested and integrated, and deep learning method can further improve the robustness and accuracy. Also, we plan to conduct a formal user study later.

VI. MANAGEMENT SUMMARY

A. Credits

Yiwei designed the software architecture of this system, implemented programs regarding AR marker tracking, registration, and pivot calibration. Modified former display calibration and UDP communication code. Integrated these with the Unity application. Tested the accuracy of four components. Also designed experiment for user study.

Mingyi conducted the hardware design of the system including adjustable camera mount and skull phantom test mount. Meanwhile, Mingyi also designed algorithm for catheter tracking, ventricle/skull segmentation and 3D-reconstruction as well as experiment for user study.

B. Deliverables

We have achieved our expected deliverables and part of the maximum deliverables include the following:

- Documentation and code for navigation system includes:
  - Anatomic points registration by tool with AR Marker in Zed mini camera
  - AR overlay system indicating ventricle centroid and catheter guidance based on anatomic points
  - User interface with workflow instruction and voice command
  - Integrated Zed mini camera system on HoloLens
  - Ventricle segmentation program on 3D Slicer
  - Zed mini camera mount design
  - Catheter tracking algorithm
- 3D-printed ventricle phantom and its mounting parts for user study
- Report of user study
- Report of AR marker tracking and tool tip tracking accuracy test
- Video demo for the workflow with final Navigation System

C. Acknowledgements

Firstly, we would like to thank Prof. Kazanzides, and Ehsan Azimi for giving us the chance to work in this interesting project, as well as their mentorship. We also thank Prof. Taylor for his guidance and advice. Last but not least, we specially thank Long Qian for his technical support with HoloLens, Unity, and 3D printing.

REFERENCES

[1] Yaniv, Ziv. "Which pivot calibration?" Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling. Vol. 9415. International Society for Optics and Photonics, 2015.

[2] Qian, L., Azimi, E., Kazanzides, P., Navab, N.: Comprehensive tracker based dis- play calibration for holographic optical see-through head-mounted display. arXiv preprint arXiv:1703.05834 (2017)

[3] R. Dockter, R. Sweet and T. Kowalewski, "A fast, low-cost, computer vision approach for tracking surgical tools," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 1984-1989. doi: 10.1109/IROS.2014.6942826

[4] Lee, Cheng Kiang et al. "Optimization of Ventricular Catheter Placement via Posterior Approaches:" Surgical Neurology 70.3 (2008): 274–277. Crossref. Web.

[5] Raabe, Clemens et al. "Revisiting the rules for freehand ventriculostomy: a virtual reality analysis." Journal of neurosurgery 128 4 (2017): 1250-1257.