CIS II Final report

Annotation Framework for Recurring Appointments in Medical Applications using Augmented Reality

Guanyu Song, May 1st
Mentor: Alejandro Martin Gomez

Part I: Technical summary

Background

Intuitive and informative medical image display methods play an important role in doctor's diagnosis and surgery process. With the increasing imaging data generated, current medical image display methods can't satisfy doctors' needs for in-situ real time display as well as managing oceans of patient personal information. So better technical solutions to combine good medical image display effects and abundant patient history information are in great demand and urgent for researchers to explore.

We want to design a system combing ultrasound tablet and augmented reality display. At first visit, doctors scan the patient body and find areas of interest, then easily save the locations and corresponding ultrasound images. For future visits, doctors can always put on a HoloLens device to see the areas annotated before and search around those locations to save the time for searching as well as call the history images for making comparations.

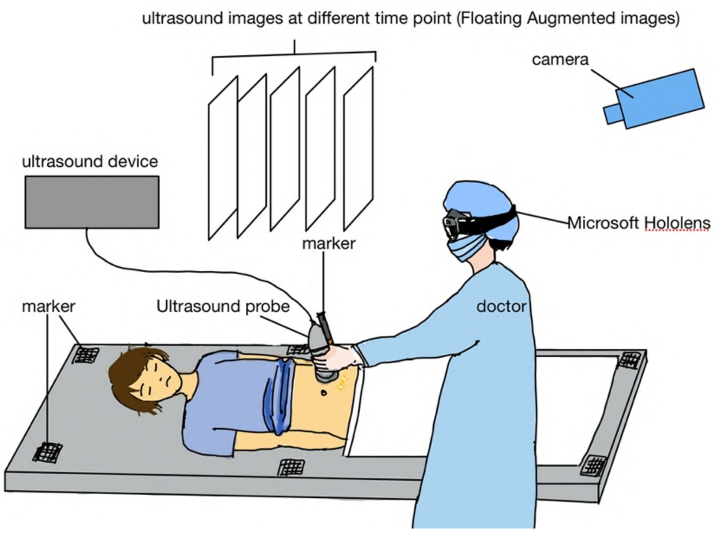


Figure1. System vision

Problem

In this project, we propose an innovative method to assist the doctors to mark locations that require special annotations during recurring appointments. Our system firstly tracks the position and orientation of the ultrasound probe with respect to the human body. In the meantime, we command the system to save the important scanning locations that are useful for the doctors and record the corresponding ultrasound images. After that, we make plans to annotate the special points saved before for future visits, so the doctor doesn't waste time for searching anatomy locations. What's more, we need to display the corresponding anatomy images captured at different time using HoloLens. All functions need to be integrated in the UI with HoloLens, so it is both intuitive and immersive for doctor's using.

Tracking, saving and augmented-reality visualization are three main problems for solving. In this class, I concentrate on tracking and saving function. Visualization process is more complicated and will be finished after this semester.

Approach

a) Tracking

We use two set of optical markers to identify the human torso and ultrasound probe model separately. Then use Vuforia Engine to track the two markers simultaneously.

In the unity Vuforia system, we can obtain the translation and rotation matrixes of the tracked markers with respect to the camera separately. And the relative translation and rotation matrixes between the markers and the two objects (ultrasound probe and human torso) are fixed. Under these conditions, we can always deduce the relative pose and position information between the probe and the phantom through tracking the markers.

Here shows the transformation relationship.

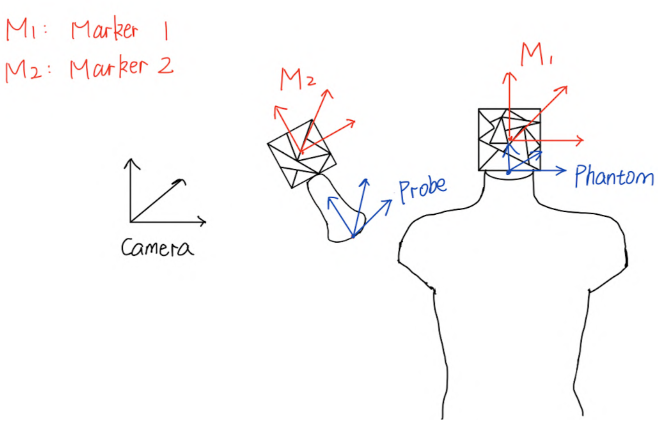


Figure 2. The coordinates transformation

From Fig 2. we can deduce that the

$$T_{phantom}^{probe} = (T_{M1}^{phantom})^{-1}(T_{camera}^{M1})^{-1}T_{camera}^{M2}T_{M2}^{probe}$$

In unity, we have lots of packages to utilize in the tracking process. We can use **Transform** to locate the position, rotation and scale of a game object. Every object in a Scene has a Transform. It's used to store and manipulate the position, rotation and scale of the object. Every Transform can have a parent, which allows us to apply position, rotation and scale hierarchically. This is the hierarchy seen in the Hierarchy pane.

Also, in Vuforia, there are different types of markers for utilizing: single image, cuboid, cylinder and 3D object. It is because that in practical application scenarios, the human body (3D object) is easy to deform, we had better choose rigid object to assist tracking. The first three types are easy to obtain and use, so I use them to track independently and test their performances.

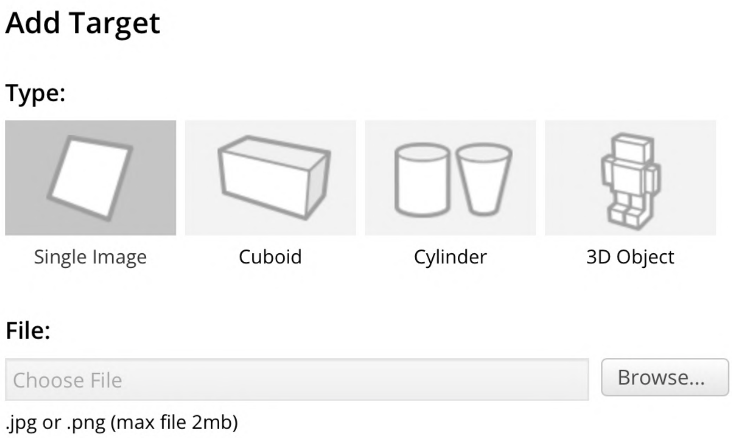


Figure 3. Different marker types in Vuforia

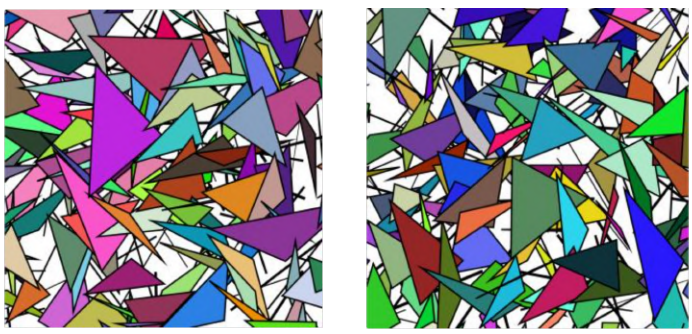


Figure 4. Sing Markers

Figure 5. 3D markers for tracking. (Left) Cube marker. (Right) Cylinder marker

At first, I employ two single markers for tracking. They are quite simple and easy to recognize for the camera, but it's obviously weak in 3D-position recognizing.

Then I change the markers as two cube targets. However, to suit the size of ultrasound probe, the cube marker attached on the probe need to be small enough (about 5cm * 5cm *5cm). But the tracking effect is quite unsatisfying. I think it is because for Vuforia recognizing, the side length of at least 10cm is required to meet the requirements for good tracking effect.

To make the 3D marker larger, my mentor advised me to use a cylinder target to track the ultrasound probe. However, it is still not sensitive enough. The marker used for tracking the ultrasound probe is quite difficult to design for the sake of balancing size and compactness, so I need further experimental research.

b) Saving

Before implementing the system to HoloLens, we use keyboard to command the computer to save the position and orientation matrixes tracked into a JSON file.

And we don't get an ultrasound device now, so we use keyboard to save the images captured by the camera aligned with the coordinates.

c) Visualization

The system plans to employ HoloLens 2 to design the user-interface and use special symbols (such as red spheres) to annotate scanning locations on the human body. By searching around the landmarks, doctors can easily find the ideal anatomical information. The inter-face will include functions to help recur the history images at the scanning locations for making comparations of the evolution process.

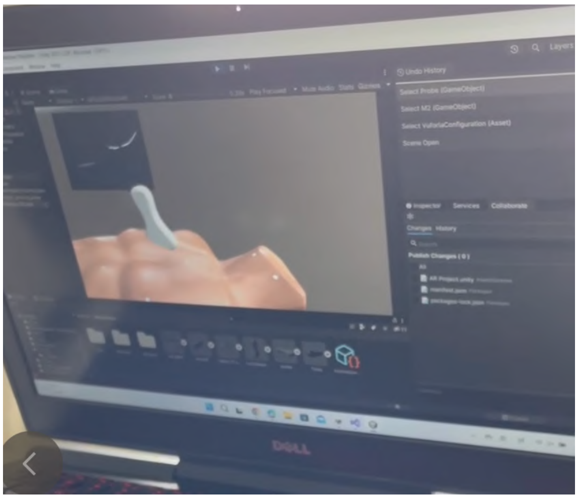


Figure 6. Application scene

Results

a) Made a 3D scanning model of the human torso for tracking.

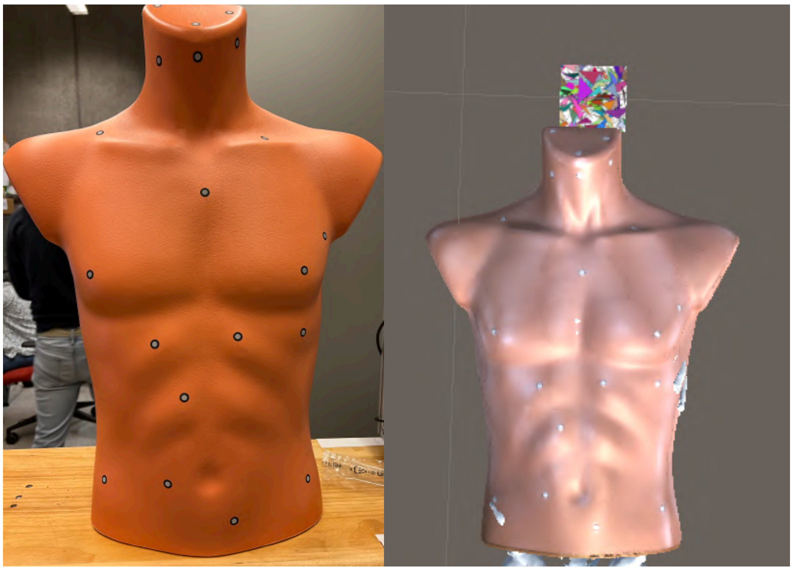


Figure 7. Torso model made for tracking

b) Make a 3D printed ultrasound model with markers for tracking.

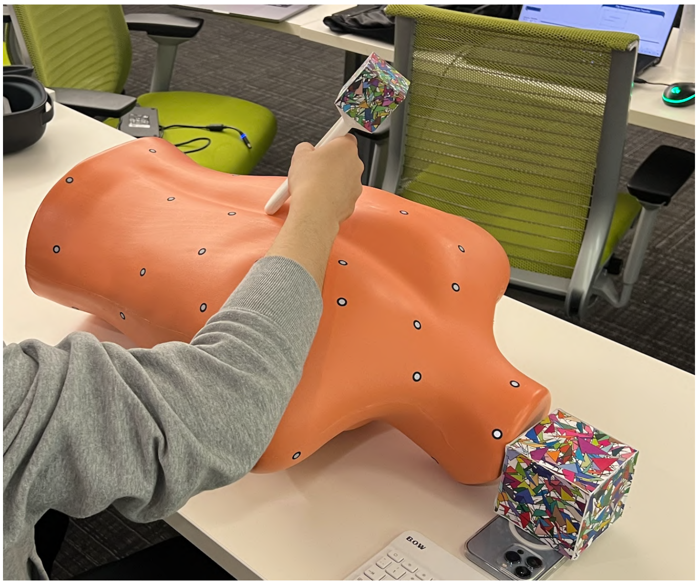


Figure 8. Testing scene

c) Track the ultrasound probe and human phantom using different optical markers. Then calculate the relative position and orientation of the markers and show the coordinates in real time.

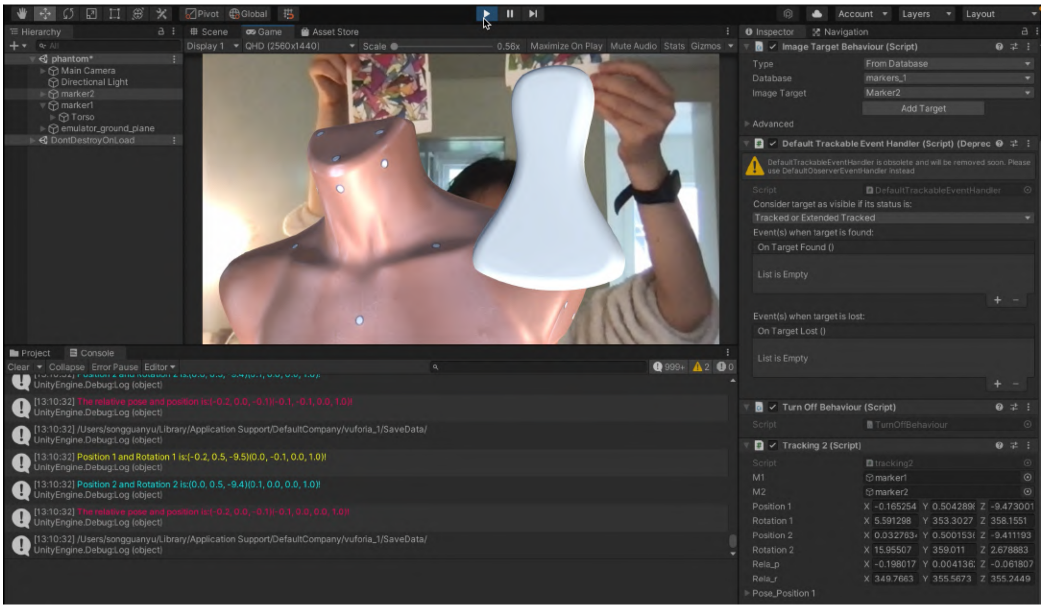


Figure 9. Tracking and estimating the pose and position in real time

d) Use space key to command the system to save the position and rotation matrixes in JSON file for future implementation.

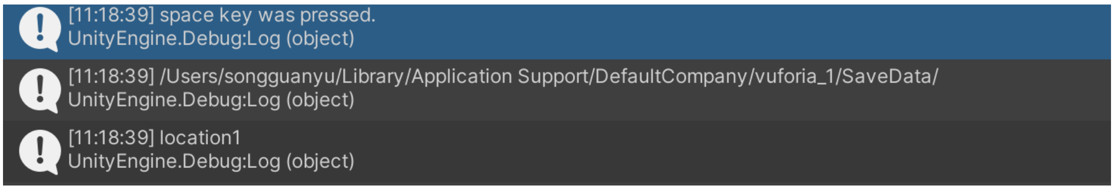
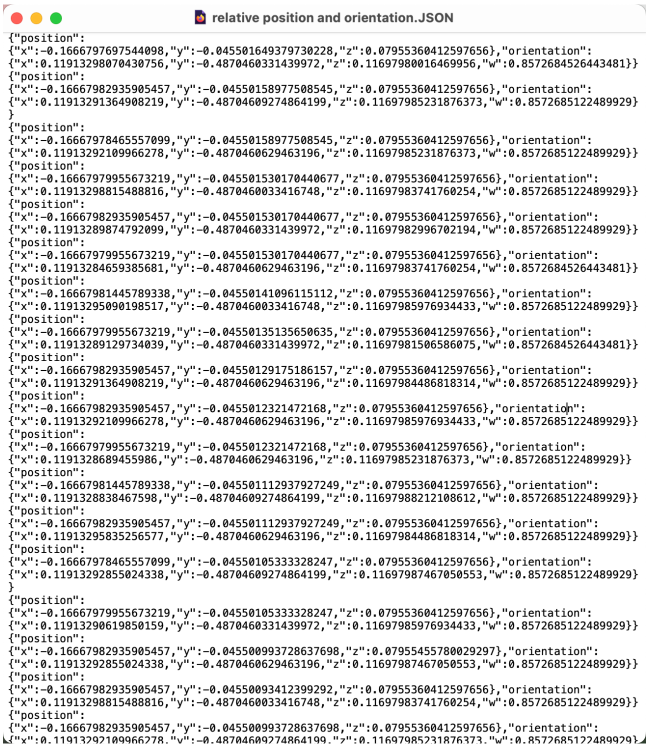


Figure 10. Use space key to save the relative pose and position of objects.



{"position":
{"x":-0.1666797697544098,"y":-0.045501649379730228,"z":0.07955360412597656},"orientation":
{"x":0.11913298070430756,"y":-0.4870460331439972,"z":0.11697980016469956,"w":0.8572684526443481}}
{"position":
{"x":-0.16667982935905457,"y":-0.04550158977508545,"z":0.07955360412597656},"orientation":
{"x":0.11913291364908219,"y":-0.48704609274864199,"z":0.11697985231876373,"w":0.857268512248992 9}
}
{"position":
{"x":-0.16667978465557099,"y":-0.04550158977508545,"z":0.07955360412597656},"orientation":
{"x":0.11913292109966278,"y":-0.4870460629463196,"z":0.11697985231876373,"w":0.8572685122489929}}
{"position":
{"x":-0.16667979955673219,"y":-0.045501530170440677,"z":0.07955360412597656},"orientation":
{"x":0.11913298815488816,"y":-0.4870460033416748,"z":0.11697983741760254,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.045501530170440677,"z":0.07955360412597656},"orientation":
{"x":0.11913289874792099,"y":-0.4870460331439972,"z":0.11697982996702194,"w":0.8572685122489929}}
{"position":
{"x":-0.16667979955673219,"y":-0.045501530170440677,"z":0.07955360412597656},"orientation":
{"x":0.11913284659385681,"y":-0.4870460629463196,"z":0.11697983741760254,"w":0.8572684526443481}}
{"position":
{"x":-0.16667981445789338,"y":-0.04550141096115112,"z":0.07955360412597656},"orientation":
{"x":0.11913295090198517,"y":-0.4870460033416748,"z":0.11697985976934433,"w":0.8572685122489929}}
{"position":
{"x":-0.16667979955673219,"y":-0.04550135135650635,"z":0.07955360412597656},"orientation":
{"x":0.11913289129734039,"y":-0.4870460331439972,"z":0.11697981506586075,"w":0.8572684526443481}}
{"position":
{"x":-0.16667982935905457,"y":-0.04550129175186157,"z":0.07955360412597656},"orientation":
{"x":0.11913291364908219,"y":-0.4870460629463196,"z":0.11697984486818314,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.0455012321472168,"z":0.07955360412597656},"orientation":
{"x":0.11913292109966278,"y":-0.4870460629463196,"z":0.11697985976934433,"w":0.8572685122489929}}
{"position":
{"x":-0.16667979955673219,"y":-0.0455012321472168,"z":0.07955360412597656},"orientation":
{"x":0.1191328689455986,"y":-0.4870460629463196,"z":0.11697985231876373,"w":0.8572685122489929}}
{"position":
{"x":-0.16667981445789338,"y":-0.045501112937927249,"z":0.07955360412597656},"orientation":
{"x":0.1191328838467598,"y":-0.48704609274864199,"z":0.11697988212108612,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.045501112937927249,"z":0.07955360412597656},"orientation":
{"x":0.11913295835256577,"y":-0.4870460629463196,"z":0.11697984486818314,"w":0.8572685122489929}}
{"position":
{"x":-0.16667978465557099,"y":-0.04550105333328247,"z":0.07955360412597656},"orientation":
{"x":0.11913292855024338,"y":-0.48704609274864199,"z":0.11697987467050553,"w":0.8572685122489929}
}
{"position":
{"x":-0.16667979955673219,"y":-0.04550105333328247,"z":0.07955360412597656},"orientation":
{"x":0.11913290619850159,"y":-0.4870460331439972,"z":0.11697985976934433,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.045500993728637698,"z":0.07955455780029297},"orientation":
{"x":0.11913292855024338,"y":-0.4870460629463196,"z":0.11697987467050553,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.04550093412399292,"z":0.07955360412597656},"orientation":
{"x":0.11913298815488816,"y":-0.4870460033416748,"z":0.11697983741760254,"w":0.8572685122489929}}
{"position":
{"x":-0.16667982935905457,"y":-0.045500993728637698,"z":0.07955360412597656},"orientation":
{"x":0.11913292109966278,"y":-0.48704609274864199,"z":0.11697985231876373,"w":0.8572685122489929}}

Figure 11. JSON file saved for future visits

e) Use S key to save the image that camera captured in real time to simulate the medical image saving scene.

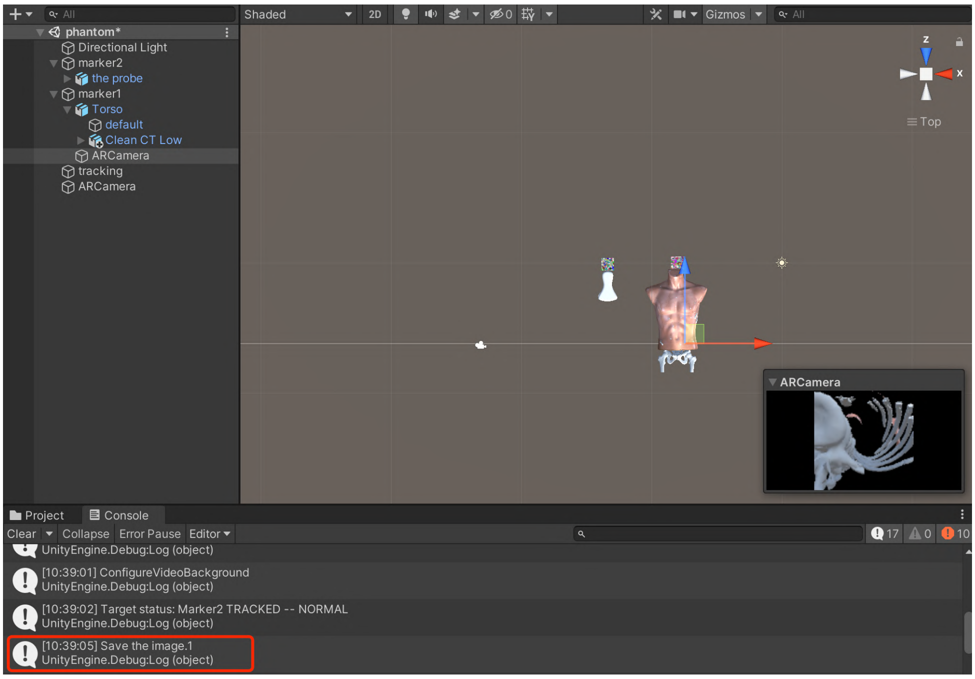


Figure 12. Save the images capatured by camera

Part II: Management summary

The models of the ultrasound probe and human phantom are made by me and my mentor. Tracking is the most important part for my overall work and it is finished by myself (but still need to be impoved). The unity packages and C# program are build by myself.

As I showed in my checkpoint presentation, here are my accomplished part (marked with red font) & planned part (marked with black font):

1. Minimum (4.15): Finishing Tracking and Saving function.
   a) **A working demo capable to track the tool pose and display visual augmentations of their pose with respect to the patient's body. Provide a git repository containing a readme file (include instructions on how to replicate, the system and its dependencies.) and the source code to replicate the project.**
      i. Save the pose and position of the tool with respect to the patient's body.
   b) **Provide c# code uploaded on google docs and github.**
      i. Record images with desired pose coordinates and show the recorded images during future visits.
   c) **Provide image files with orientation and position information.**

2. Expected (4.30): Testing on a human body phantom.
   a) In addition to the minimum, it is expected to include a 3D scanned replica of human body phantom and a 3D printed ultrasound probe.
      i. Make 3D print of ultrasound probe model and use 3D scanner to create a virtual

replica of the phantom.
Provide 3D reconstruction file.

ii. Evaluate the accuracy of the tracking technique.
Provide a report containing the results of the tracking accuracy

3. Maximum (5.1-): Adding body tracking function.
   a) Automatically identify the patient's body pose. Visualize the human body anatomy evolution with HoloLens display.
      Provide source code for the tracking of patient's body pose.

What to do next

1. Improve the marker for tracking ultrasound probe model so we can get better tracking results.
2. Record a real-time tracking demo and **estimate the accuracy.**
3. Deploy the visualization system to HoloLens 2 device and make a complete medical display and diagnosis system.

What I have learned

To finish this project is really an exciting experience and provide me a fresh perspective to use augmented reality technique to solve medical problems. I have learned quite a lot trivial but important skills,

1. The essentials of building unity projects.
2. Methods and tips for employing Vuforia Engine.
3. Start programming with C#.
4. Experiences on perform specific experiments.

The process of applying research is a construction process from existence to non-existence. We always meet problems and find different ways to solve them. In this process of solving problems, our mentality is exercised and our problem-solving skills are strengthened step by step. It's a unfamiliar, challenging but excellent trip.

Part III: Technical appendices

Provided on my Git: https://github.com/stefaniesong/Ultrasound-AR-tool-tracking-system
(including unity packages, C# codes, readme file and other resources)

References

1. El-Hariri H, Pandey P, Hodgson A J, et al. Augmented reality visualisation for orthopaedic surgical guidance with pre-and intra-operative multimodal image data fusion[J]. Healthcare Technology Letters, 2018, 5(5): 189-193.
2. https://library.vuforia.com/features/images/image-targets.html
3. https://docs.unity3d.com/ScriptReference/Transform.html
4. https://docs.unity3d.com/ScriptReference/KeyCode.html