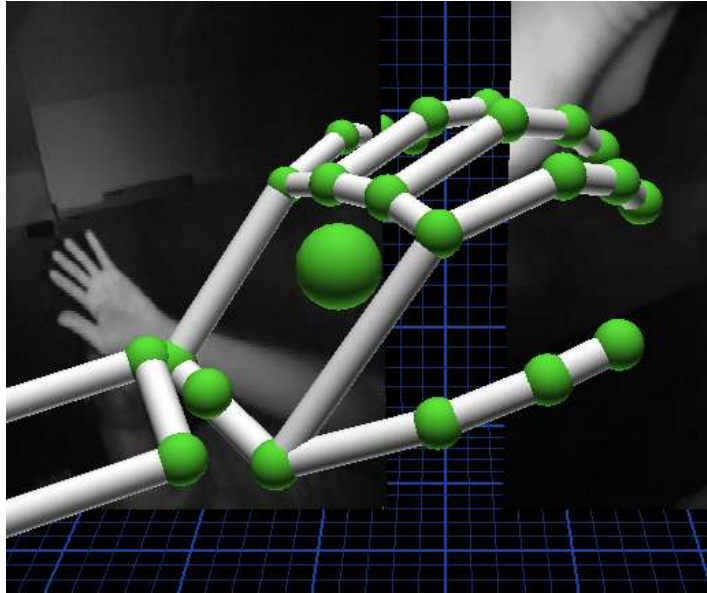


Leap Motion Controller

Leap Motion controller was used to track a maximum number of two hands in 3D space.



It can track the hand along with fingers and joints. Also, it can track some tools like a pencil which the user can hold in the hand. The device employs two infrared cameras and three infrared LEDs to track the hand motions. The wavelength of infrared used is 850nm and it is out of the range of visible light spectrum. The volume which can be detected is 2 feet above the device and 2 feet wide on either side. The images captured by the cameras are greyscale stereo of near-infrared light spectrum. Using these images, the position and orientation of hands are determined. The device communicates with the computer via USB interface.

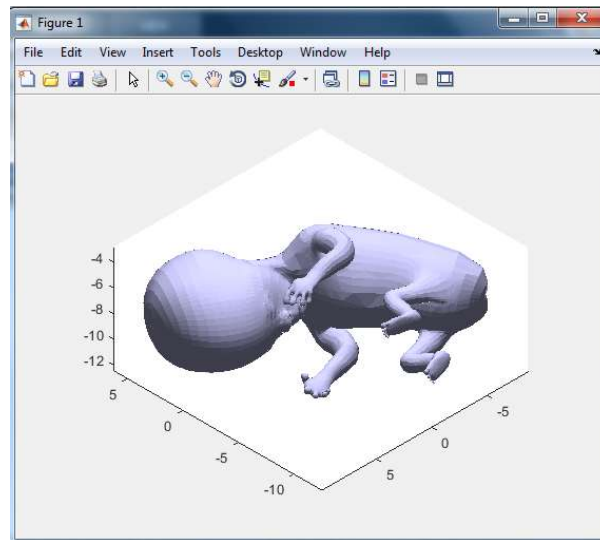
The data from the Leap Motion tracker can be read with the help of Leap Motion SDK. In this project, Leap Motion SDK (v.2.3.1.31549) was used. Leap Motion provides access to its SDK via JavaScript, Unity/C#, C++, Java, Python, Matlab and Objective-C. For this project, Matlab is used. Leap Motion website provides header files and sample codes required for this.

Mesh Files

Two mesh files are being used in this project at the moment. One mesh file represents the fetus to be displayed in the ultrasound simulator. It is in .stl format, which is widely used to represent meshes. The mesh contains around 79,000 points and triangular faces. Another .stl mesh representing the heart is also being used in this project. These meshes can be visualised in the software called MeshLab.

Matlab

To import the .stl file into Matlab environment, a Matlab program called stlRead.m created by Eric Johnson is used (http://uk.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader/all_files).



In ultrasound imaging of fetus, cross sections of the fetus should be displayed on screen. The mesh is made up of coordinate points in 3D space and faces. After reading the .stl file in Matlab, an array of coordinate points and an array of points which make up the faces are obtained.

To display a cross section of the fetus with above arrays as inputs, a plane was defined in Matlab and points crossing that plain were obtained. When that plain passes through a triangular surface, the point of contact was obtained. Afterwards, the coordinate points of these points are stored in an array.

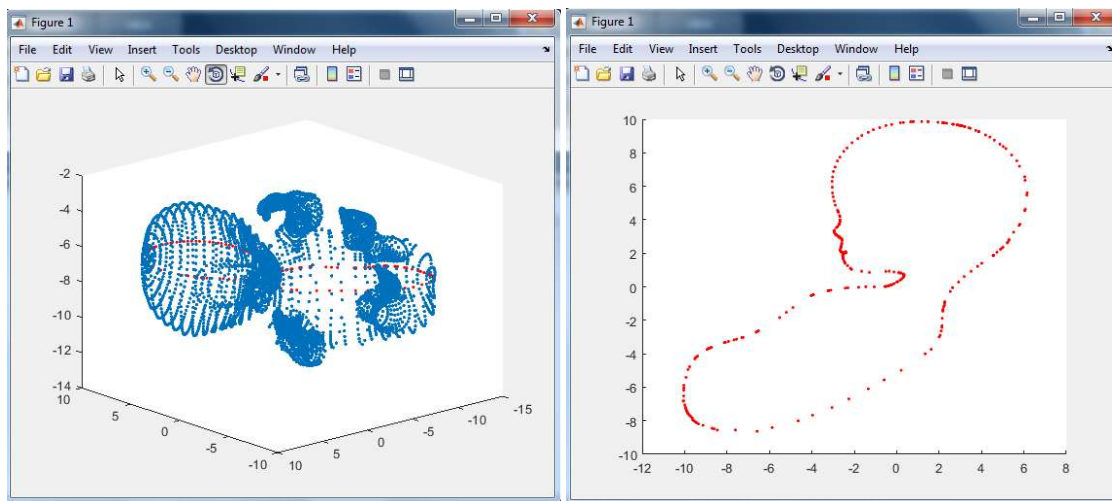
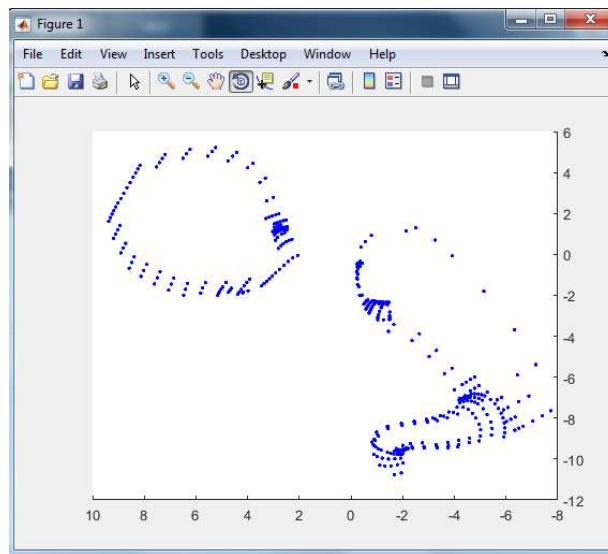


Figure xx shows the plane of cross section along with the point cloud. Figure xx shows the normal view of the points obtained.

However, the code takes a considerable amount of time to show the figures. Therefore, it will be not suitable for showing the ultrasound image in near real time. Time taken to do the calculation and show the updated image is usually around 1.5 seconds.

To avoid this problem, a simpler algorithm was suggested. It simply calculates the distance for each point from the surface defined and takes the points with distances less than some threshold value.



This slightly improves the performance. Now the time required to show the updated image is around 1.0 second. Still, the processing time is too long for near real time rendering of ultrasound images.

Therefore, one of the next goals is to find an efficient algorithm to do the calculation and show images.