

# Tracking Objects on a Deformable Surface using displacement and orientation data

Zachary DeStefano, Kyle Culter, Gopi Meenakshisundaram, Bruce Tromberg  
University of California, Irvine

August 13, 2014

## Abstract

Previous attempts at tracking objects as they move have used magnetic fields or a camera array. With deformable surfaces trying to track on them is especially difficult because the path is not directly following the original shape for the mesh. For this paper, we present an algorithm for tracking on a deformable surface as well as a way of tracking using only displacement and orientation sensors. We developed a probe that contained an optical mouse sensor for displacement on the surface and an accelerometer, gyroscope, and compass to find out the orientation of each displacement. We used this data to record paths and developed an algorithm for doing calibrations as well as following the surface.

## Introduction

In various medical applications, there is a need to track a probe as it moves across a patient's body and takes measurements. Currently they attach a grid to the patient and painstakingly take measurements at each point in the grid. This led us to explore ways of automating that process. Using a Kinect to try and track probe movement can be impractical because the probe needs to always be in view of the Kinect camera. Magnetic grids are expensive and cumbersome so that option would not have worked well. We then thought that using an ordinary optical mouse sensor on the surface itself could work well. Because the surface is not flat, we will need to know the orientation so we put a gyroscope, compass, and accelerometer into the probe. Using these simple tools as well as software that implemented the calibration algorithm shown in this paper, we were able to get a reasonable approximation of the probe's path on the surface itself.

Our process started with taking a 3D scan of the surface to get the mesh. We use a Kinect to get the data and the point cloud architecture **\*\*INSERT MORE DETAIL\*\*** to take the data and get a mesh and texture from it. We then take out some of the noise and use the QSlim **\*\*INSERT MORE DETAIL\*\*** algorithm to simplify the mesh. Once the mesh is simplified, we load it into our tracking environment. Our tracking environment is an application that takes in the mesh as well as probe data readings and gives live updates as to the probe's position and data. It operates similarly to a video game and was developed using JMonkeyEngine **\*\*INSERT DETAIL\*\***, a video game library written in Java that is similar to Ogre **\*\*INSERT REFERENCE\*\***.

The first step is to make sure the scale is calibrated. This is done in two steps. We move the probe a certain distance and see what the total displacement outputted by the mouse sensor was. We then measure the distance between known points on the surface in the virtual world and the real world and see what the ratio is. We then combine the two ratios so that we have a scale factor to go from displacement units read by the probe to displacement units in the virtual world. Once the scale is calibrated, we start recording paths on the mesh itself. We then have a problem. Because the rotation has not been calibrated, it is likely that the paths will have a shape similar to the part of the mesh they were on but they will not actually be on the mesh **\*\*INSERT PICTURE OF THIS\*\***. We thus need to figure out what rotation to apply to the probe reading's rotation in order to get the rotation in the virtual world. After that though, there is still a problem. Because

the surface is deformable, the path would still not perfectly follow the mesh. We thus need an algorithm that will approximate the probe's position on the mesh given the deformability.

This paper presents two algorithms. The first one takes a path that almost follows the mesh and projects it onto the mesh in such a way that preserves its orientation along the mesh as well as its arc length. The second one takes a recorded 3D path between two known points on the mesh and figures out how to rotate it and project it onto the mesh so that its start and end points match the ones on the mesh. The rotation found by the second algorithm gives us the rotation calibration we need. Once the rotation is calibration is found, we can then live track the probe and use the first algorithm to keep it along the mesh.

## **Related Work**

Surface flattening paper entitled mesh unfolding

Magnetic tracking:

<http://s2014.siggraph.org/attendees/emerging-technologies/events/im3d-magnetic-motion-tracking/>

## **Following a Deformable mesh**

## **Calibration of Rotation**

## **Experimental Results**

## **Conclusion and Future Work**

## **Acknowledgments**

## **References**

## **Appendix**