

## Problem Set 6

### Haptic Rendering of Simple Dynamic Environment

Out: 10.24.06

Due: 11.01.06

Using the Phantom Omni, the OpenHaptics API, and OpenGL, create a virtual environment with the following properties:

1. (20 pts.) A large square box surrounding the user's haptic interaction point. The box should be haptically and graphically rendered. Select a reasonably high stiffness and wall positions that allow plenty of room to move around inside. Do not graphically render the wall closest to the camera, so that the user can see inside. The user should be a very small sphere. The haptic rendering should take into account the radius of the user sphere, and visually should not show the user penetrating into the walls.
2. (10 pts.) Create a "large" sphere that resides in the box. The large sphere should be haptically and graphically rendered. The sphere diameter should be approximately one-fifth of the width of the box, and much bigger than the user's sphere. The surface of the large sphere should be smooth and frictionless. The haptic rendering should take into account the radius of the user sphere, and visually should not show the user penetrating into the large sphere. I suggest that you start with a stationary sphere before going on to the next step.
3. (20 pts.) Add dynamics to the large sphere, so that it moves around inside the box when the user pushes on it and bounces off the walls. There should be viscous damping between the sphere and the environment, so that it will eventually slow to a stop if the user does not push on it. It should have zero initial velocity. You should select the mass, stiffness, and damping properties to provide a pleasing, intuitive virtual environment for the user. You can make the interaction between the sphere and the box walls a perfect reflection (with no penetration), or you can generate forces between the sphere and the box walls due to penetration.

You should begin with the C:\Program Files\SensAble\3DTouch\examples\HD\graphics\CoulombField code that comes with the OpenHaptics API, copy it to your own directory, and edit only three files, called "helper.h", "helper.cpp", and "main.cpp". For submission, rename your final files to be called "helper\_PS6\_YOURLASTNAME.h", "helper\_PS6\_YOURLASTNAME.cpp", and "main\_PS6\_YOURLASTNAME.cpp". Combined, these files should create a single virtual environment that satisfies all three properties described above. Turn in hard copies of these files and also email them to david.grow@jhu.edu.