Programming Assignment #2

Due Date: Tuesday, April 22, 11:59 PM

You will be implementing a proxy-based haptic rendering algorithm in this assignment in the form of a renderer for implicit surfaces. You will also see how a simple surface effect, friction, can be added to the rendering algorithm.

This assignment is worth a total of 15 points, with an addition possiblity of earning a "bonus" designation. Since the bonus is a binary state, it will only be awarded to submissions that do an exemplary job of meeting the requirements.

Getting Started

By this point, we hope that you have obtained a Novint Falcon haptic device and have successfully got it working with the CHAI3D library. Start by copying and renaming the "application" template located in the "templates" folder of the CHAI3D package. Download the additional starting file archive (CS277-Assignment2-Code.zip) from the course web site and unpack it into your local folder. You will need to replace application.cpp with the one provided, then add ImplicitMesh.cpp/h and MarchingSource.cpp/h to the Visual Studio or Xcode project as well.

There are a few more files in this package than in the previous homework, but that's no cause for concern. The main function is again in application.cpp, which also sets up the scene graph with an interaction cursor and a solitary object. However, most of your work will be done in the ImplicitMesh class, as this is where the rendering algorithm should be implemented. The MarchingSource files contains a public domain implementation of the marching cubes algorithm that we use to polygonize the implicit surface for visual rendering. You can safely ignore those source files.

Part I - Rendering Implicit Surfaces (8 points)

Use the implicit surface rendering algorithm discussed in lecture to render four different implicit surfaces: a sphere, a heart, a whiffle cube, and one of your own choosing. The objects and their respective implicit functions are shown below.

Your algorithm can be implemented in the computeLocalInteraction() method of the ImplicitMesh class. The rest of the program has already been set up so that when you complete that function, you will be able to interact with the object and feel forces as expected. Add to your program a means for switching between the four different surfaces (using the keyboard, a button on the Falcon, etc.).

When you render the heart shape, run the device along the depression on the top of the shape – you'll likely notice the limit cycle effect discussed in lecture. It manifests as a buzzing or rapid vibration. You will need to alter the basic algorithm to address this problem for full credit on this part. Some ideas for this are discussed near the end of the article describing the algorithm (Salisbury & Tarr 1997, available on the web site).
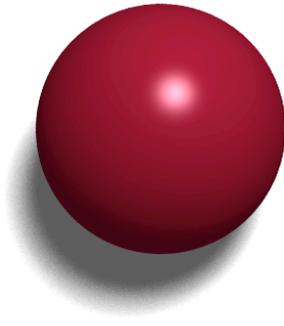
Sphere
$$x^2 + y^2 + z^2 - 1 = 0$$

Heart
$$(2x^2 + y^2 + z^2 - 1)^3 - (0.1x^2 + y^2)z^3 = 0$$

Whiffle Cube
$$\left(x^8 + y^8 + z^8\right)^8 + \left(x^2 + y^2 + z^2 - 0.44\right)^{-8} - 1 = 0$$

Your Surface
$$S(x, y, z) = 0$$

Notes & Hints:

‣ The visual rendering provided in the template is only an approximation of the actual implicit surface – it should actually feel smooth despite appearing faceted. If the visual update is slow on your computer, increase the granularity parameter when the surface is created.

‣ Try to find or invent an interesting surface for your fourth object. Variations on spheres and ellipsoids will not be looked upon favorably...

Part II - Rendering Friction (4 points)

Add the capability for rendering both static and kinetic varieties of Coulomb friction to your implicit surface algorithm. Provide a means for turning the friction on and off, and for adjusting the coefficients of friction. It can be as simple as a way to toggle between no friction, low friction, and high friction.

Bonus Part - Multiple Objects

This assignment only asks you to render a single implicit surface at a time, but can you render more? Try adding an additional object to the scene graph. What happens?

Modify your implementation of the implicit surface rendering algorithm so that it can correctly render multiple (possibly overlapping) implicit surfaces in the same scene. The rendering should still maintain all the desirable properties of proxy-based haptic rendering. For example, in the scene depicted below, though each sphere is a separate object defined by its own implicit function, you should not be able to push the proxy through the center where the four spheres meet.
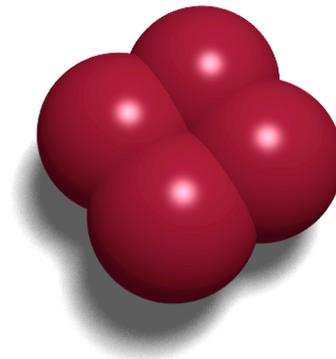
Assignment Questions (3 points)

These questions help you think about various aspects of implementing the assignment, and it may be helpful to read through and think about them before starting on your code. You may include a separate file in your submission with your answers, or simply append your answers to your "readme" file.

1. Can you think of an implicit function or class of implicit surfaces that this algorithm cannot render? Why would this surface cause problems?

2. In Part I, briefly describe why you think the limit cycle occurs at the tip of the heart shape, and what you did to address this problem.

3. In Part II, trying to determine exactly when to switch from static to kinetic friction, and vice versa, can be a tricky task. Briefly describe the conditions under which you switch from one type of friction to the other. Keep in mind that when performing calculations with floating-point numbers, a small amount of numeric precision error can be introduced. Does this pose a challenge?

Submitting the Assignment

You should normally just submit the source code for any file(s) you modified or created. Please do not send us back the CHAI3D libraries or any other large compiled binaries! We will compile your program with a pristine copy of CHAI3D when we grade your assignment.

Include a "readme" file with your submission. Please indicate which platform and development environment you used to do the assignment, in case we run into any problems compiling or running your program. Also give brief instructions on how to use your software, and any gotchas we may encounter while grading it. You may also write in this document anything else you'd like us to know about your submission.

Completed assignments should be emailed as attachments to cs277.2014@gmail.com before 11:59 PM of the due date. Please indicate any late days used. If your files are abnormally large, you may send a web download link, drop off a copy on a flash drive, or make a suitable alternate arrangement.

References

Salisbury, K. & Tarr, C. Haptic rendering of surfaces defined by implicit functions. In *Proc. Haptics Symposium,* 61-67 (1997).