Haptic modeling of a street intersection using the Novint Falcon

Rebecca Mason
mason20r@mtholyoke.edu

Dr. Roberto Manduchi (Advisor)
manduchi@soe.ucsc.edu

Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064

1. Abstract

We examined a new haptic device, the Novint Falcon, to see if it could be used for a haptic map application for the blind. To test its capabilities, we developed a haptic model of a four-way street intersection for the Falcon. We were able to accurately model surfaces, edges, and grooves using a free program and scripting language, GlovePIE. We conclude that the Falcon would be a suitable device for this task, although developing the interface between the Falcon and the application would be a challenge.

2. Introduction

The ability to travel safely and efficiently is a challenge for persons with visual impairments. Blind people cannot utilize visual aids such as landmarks and street signs, and may be unaware of hazards in an environment. Tactile maps suffer from problems with Braille labeling and feature annotation, and are therefore less detailed than their traditional counterparts. However, a haptic virtual map can convey more information because it isn't limited to what is physically possible with paper or plastic.

Haptics is a growing field focused on technology that interacts with the user's sense of touch. A haptic device sends forces, vibrations, and motions to the user, who interprets them as surfaces, shapes, and textures. Previous work in the design and implementation of haptic graphics for the blind [1, 2, 3] mainly have been done using the Phantom Desktop, produced by SensAble Technologies. [4] However, the Phantom costs over $10,000, which makes any application developed for it unlikely to be adopted by blind users.

Our project explores the use of a new haptic device, the Falcon, which was produced by Novint for the mass market. The Falcon costs under $200, which would make it affordable for blind users. We developed a simple model for the Novint Falcon to test if it is suitable to use for a virtual map application.

3. Tools

i. Novint Falcon

The Novint Falcon [5] is a three degree-of-freedom haptic device that allows the user to feel virtual objects. The Falcon is a parallel robot with three servo arms connected to a detachable grip. The size of the 3D workspace is 4 in. (10.16 cm) in each direction[1], and the Falcon can measure its position with a resolution of over 400 dpi.

Figure 1: Front view of the Falcon

---

[1] Size according to the specifications given by Novint. In practice, the Falcon's workspace seems to be about 11 cm in the X and Y directions, and between 12.5 and 13 cm in the Z direction.

This information is sent via USB 2.0 to the haptic application, which calculates the appropriate force vector to be sent to the user. The force is generated by updating currents to the motors in each arm, and felt by the user through the grip. The maximum force of the Falcon is over 2 lbs. (8.9 N). This cycle occurs at a rate of 1000Hz.

Figure 2: The Falcon in use.

ii. HDAL SDK

The Haptic Device Abstraction Layer (HDAL) is Novint's programming interface for the Falcon. The current version (v2.0.0) was released in March 2008. The SDK is for C++ and is Windows only. Once HDAL is initialized in an application, it starts a separate servo thread to manage communication with the Falcon. The servo thread uses callback functions defined in the application to calculate forces and send them to the device. The application programmer is spared from most device specific details and thread safety issues.

iii. GlovePIE

GlovePIE [6] is a freeware input emulator, capable of interfacing with a variety of devices such as MIDI controllers, Nintendo's Wiimote, and the Novint Falcon. It also has its own scripting language, which runs the same code snippet continuously until the user exits the program. Although GlovePIE originally wasn't meant to run fast enough to be used with the Falcon, the frame rate can be set to sufficiently high values, and running on a decent computer there didn't seem to be any problems.

4. Haptic Modeling of a Solid Surface

When one presses a finger hard against a solid surface, the skin and muscles of the fingertip compress, causing a feeling of pressure. As the finger moves away from the surface, the skin and muscles return to equilibrium and the feeling of pressure goes away. The force of the finger pushing against the surface is proportional to the normal force of the surface pushing back.

However, most haptic devices don't measure force, they measure position. So to create an illusion of a solid object, we model its surface as a spring. The user pushes through the surface, and a restoring force pushes the user back towards equilibrium. The formula used to calculate the force vector is based on Hooke's Law:

$$\mathbf{F} = -k\mathbf{x}$$

where k is the surface's spring constant in newtons per meter, $\mathbf{x}$ is the distance in meters between the user's position and the surface, and $\mathbf{F}$ is the restoring force exerted, measured in newtons.

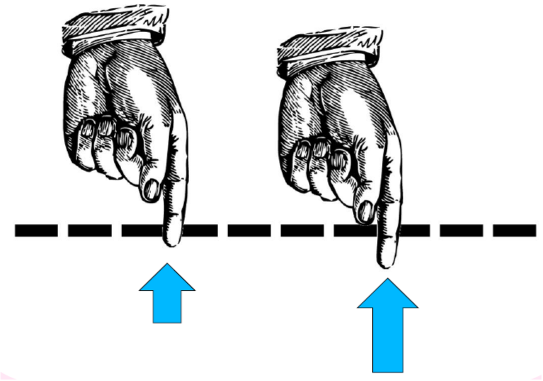

Figure 3: The magnitude of the force exerted by the Falcon increases proportional to the distance the user pushes through the surface.

There is no one spring constant for solid surfaces that works across all simulations. In general, low spring constants make solid objects feel soft and spongy, while high spring constants make objects feel rubbery and sticky. High spring constants also make it difficult for the user to feel along the sides of objects, which is a problem for blind users. Also, if two surfaces with high spring constants face each other, the device could start bouncing between them, causing violent oscillations.

5. Model Design

The test model we made for the Falcon is a haptic representation of a four-way intersection. The intersection is modeled as two crossing perpendicular grooves cut into a flat surface. Here are some of the things we considered while making this design:

- **Orientation**

Early models placed the surface of the map on xz-plane, partially because it seemed more realistic to do so. Later, we rotated the map so that the surface was on the xy-plane, facing the user's fingertips.

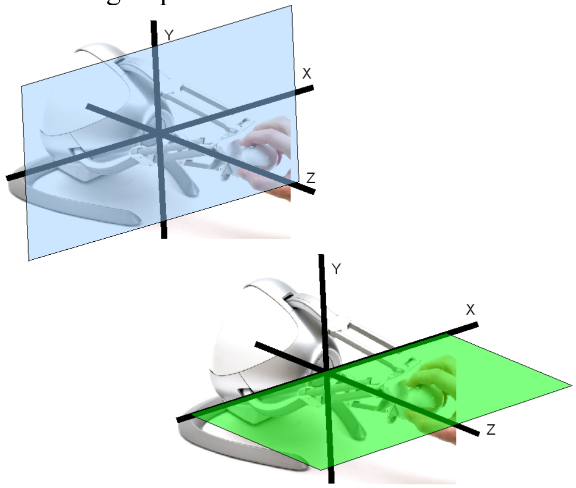

Figure 4: Top, xy-plane. Bottom, xz-plane.

Originally this was done because it seemed to make the streets easier to follow, but we also discovered that this drastically cut down the amount of oscillation because the device could not longer oscillate in the z-direction, where the force exerted by the user tends to be greatest.

- **Streets as Grooves**

The reason why the streets in this simulation are modeled as rectangular grooves is because of research by Ramloll *et al.* which found that blind users have a much easier time following virtual grooves than following virtual raised lines. [1]

- **Free vs. Restricted Movement**

While defining the size of the rectangular groove, one concern we had was that the user might feel “lost” in a groove that is too deep. After some experimentation, we found that deep grooves were not a problem because there isn't any need for users to be able to follow edges at the map's surface, and because no matter how deep they are, the users can always rise back to the surface freely.

However, if the groove is too wide or too shallow, that does become a problem. Users might get confused or lost because they can't tell if they are in a groove or on the surface of the map. At the moment, our best intersection model has grooves that are four times as deep as they are wide.

Blind users can feel overwhelmed while freely navigating haptic environments. Often they have an easier time building a spatial representation if their sense of touch is guided or restricted only to things that are important. Our two-layered model allows users to move freely above the surface of the map, and then to “push in” into narrower grooves in order to follow a route or feel a curve in a road.

6. States and Force Calculations

The 3-d model is split into discrete areas. There are four types of area where the user can move freely: the area above the map, the area inside the vertical street, the area inside the horizontal street, and the area where the streets intersect.

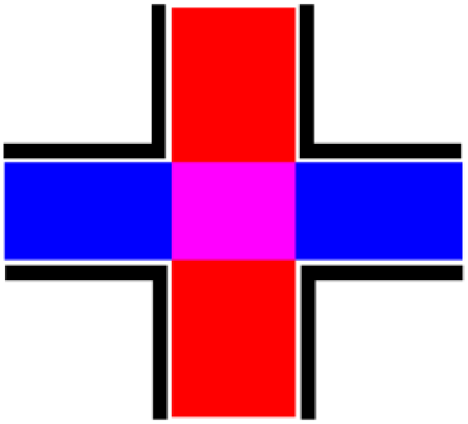

Figure 5: Looking at the lower layer of the model from the top-down. The colored areas are where the user can move freely, while the white areas represent “solid” areas.

The user's state changes as the user moves from one free area to another. The state does not change when the user moves from a free area to a solid area.

The user feels “solid” objects because of the spring force pushing the user out of the “solid” areas. These forces are calculated by pushing the user back in the direction from which the user entered that area. This way, if the solid area is between two free areas, the user will not be able to push into one free area from another.

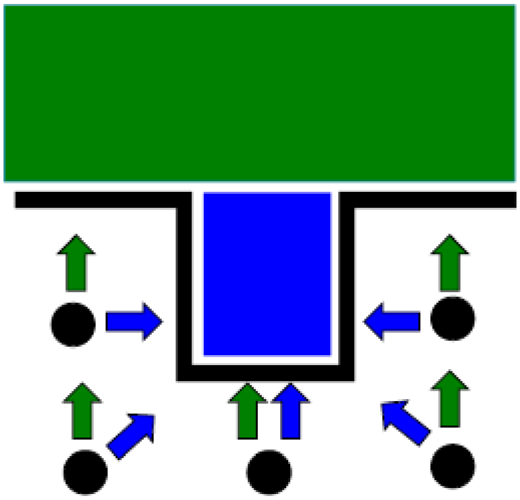

Figure 5: A cross-section of the model, which includes the inside of a groove (in blue) and the free area above the maps surface (in green).

The diagram above shows a cross section of the model, with the arrows representing the direction of the force that is applied depending on the user's state. As the diagram shows, the user is always pushed back into the free space that the user entered from.

7. Results

The original plan was to create the test models using HDAL and C++, but we ended up switching to GlovePIE after about two weeks of struggling with the HDAL API.

For instance, the SDK example of a simple cube involves defining all 12 inner and outer surfaces, using matrix math to determine which surface is closest to the user's position, and then calculating each of the component vectors of the restoring force.

It took us a lot of work in order to find the optimal spring constants for the model. These are the constants that we used:

For the following groove size:
Horizontal groove width = 0.6 cm
Vertical groove width = 0.6 cm
Depth of grooves = 2.4 cm

We used these spring constants:
Spring constant for X and Y directions
= 15.38 Newtons/meter
Spring constant for Z direction
= 45.45 Newtons/meter

The reason why the spring constant for calculating forces in the Z direction is higher is because we noticed that users tend to push the device harder towards and away from themselves than they do up and down or from side to side. Using a higher spring constant in the Z direction allows us to make the model as stiff as possible while still avoiding oscillations in the X and Y directions.

8. Further Work

Further work needs to be done to increase the amount of information that can be conveyed by the model. Adding different textures to surfaces of the map could quickly indicate different types of areas the way that colors are used on maps for sighted people. Information about elevation could also be added to the map by making the grooves deeper when the elevation is lower. Then the user would be able to feel the change of the elevation in the road as they feel along it on the map.

There are a few ways in which the Novint Falcon could be used in an actual map application. While the GlovePIE scripting language isn't fast enough for a more detailed model, the Open Sound Control protocol could be used to have a map application communicate with the Novint Falcon. The map application would have to be very fast though, in order to calculate the forces at 1000 times per second.

We recently learned about HAPI, which is an open-source, cross-platform, haptics rendering engine, which is compatible with the Novint Falcon [7]. It is possible to model a 3-dimensional model using OpenGL and then use

HAPI to create the haptic model. This would make it very easy to create larger and more detailed models automatically for a real-life application, using outside information from GIS or Google Maps.

9. References

[1] W. Yu, R. Ramloll, and S. Brewster, 2000. Haptic Graphs for Blind Computer Users. Lecture Notes in Computer Science, 2058 pp. 41-51

[2] C. Springsguth and G. Weber, 2003. Design Issues of Relief Maps for Haptic Displays. Available at: http://www.multireader.org/design.htm last accessed 18 August 2008.

[3] G. Nikolakis, D. Tzovaras, S. Moustakidis, and M. Strintzis, 2004. CyberGrasp and PHANTOM Integration: Enhanced Haptic Access for Visually Impaired Users. Available at: http://www.isca-speech.org/archive/specom_04/spc4_507.pdf last accessed 18 August 2008.

[4] http://www.sensable.com/

[5] http://home.novint.com/

[6] http://carl.kenner.googlepages.com/glovepie

[7] http://www.h3dapi.org/modules/mediawiki/index.php/What_you_can_do_with_HAPI