Haptic Rendering of Large-Scale VEs

Dr. Mashhuda Glencross and Prof. Roger Hubbold – Manchester University (UK)
Perceiving the Sense of Touch

Important considerations:

• Burdea: Haptic response within 1KHz or better
• Severe limits on computation
• Devices types vary
  – Impedance Control.
  – Admittance Control.
The FCS HapticMASTER

Device used:
- 3 DOF input 3 DOF output
- Admittance control
- Server PC controls arm
- VxWorks real-time OS
- FCS haptic API

Photo courtesy of FCS
The FCS HapticMASTER

Device workspace:

Image courtesy of FCS
Vendor APIs are never enough:

- Researchers always want to do more than the API supports
- Building haptically enabled apps should be easy
- Application developer should not have to be aware of the needs/limitations of the haptic rendering
- Developers should not have to treat graphical/haptic representations separately
A VR system with Haptics:

- Graphical primitives manage haptic counterparts
- Transparent complexity management
- Accelerated Graphics
- Spatial partitioning
- Platform independent
An extendable framework:

- Maverik++ - OpenGL based
- Both Maverik++ and Cohap3D are C++ libraries
- Cohap3D transparently uses the FCS haptic API to program the haptic server
Cohap3D – Case Study

Haptic rendering of Large-Scale VEs:

• Haptic server handles limited number of primitives
• Automatic culling to a haptic region
• Cache mechanism takes candidates and chooses those to activate/de-activate
• Continually updates as the user navigates
Case Study – Caching policy

Overview:

Voxellised region of space

Possible candidates

Enabled
Large-Scale Models

Video showing cached primitives added and removed as the user navigates:
Large-Scale Models

Video showing a user haptically interacting with a large model:
Current issues which affect usability:

• Primitives may be enabled after the user has accidentally passed through them
• The application may start up with the end-effector inside a haptically enabled primitive
• Very small thin primitives may not offer much resistance
• Not all the primitives in our models are currently supported
Case Study – Performance

Performance:

• Number of primitives - > 25,000
• Polygon count - > half a million
• Average number of possible candidates - 300
• Haptically enabled primitives – 30
• Client PC spec – 3.2 GHz, 1GB RAM, NVidia GeForce 5950 FX
• Haptic server PC spec – low by modern standards
Additional Cues

Model scale vs. device scale:

• Model scale is very large

• Operating scale of device is small
  – Only a small portion within reach

• Colour mask applied to everything outside the workspace
  – Improves user’s ability to navigate/interact
Navigation

• We needed buttons!
  – Built our own

• Navigation scheme
  – Appears as a standard mouse
  – Similar to a rate controlled joystick
Limitations

Most Frameworks don’t do everything:

• Currently only supports a HapticMASTER device
• Haptically enabled primitives need to be generalised
• Currently only allows single user haptically enabled VE applications to be built
Future Developments

- Improve culling algorithms
- Extend API for Phantom
- Support collaboration
Further Information

Further information about the research carried out by our group can be found at:

http://aig.cs.man.ac.uk