Chapter 10

Module 8

Visualization

- In visualization, we are concerned with *exploration*
- In computer-graphics, we are concerned with *rendering*

The use of 3D

- Analog with *real-world* physics.
- 10-fold improvement in item density with 3D.
- Familiarity with spatial location helps reduce visual clutter.
- Need sufficient visual cues.

Use of 3D
OpenGL

✔ SGI in-house graphics system
✔ Now a widely accepted graphics standard
✔ Standard on UNIX and Windows
✔ API supports rendering, buffering, anti-aliasing, shading, colouring, texture-mapping, a display list, Z-buffering...

OpenGL Application

Java3D & VTK

✔ 3D OO toolkits
✔ VTK is open source
✔ C++ class library, and
✔ interface layers for Tcl/Tk, Java, and Python.
Network traffic application

To help answer questions such as the following:
- Which segments carry the most traffic?
- Which sections of the network are down?
- At what times, and where do traffic bottlenecks occur?
- ...

Application elements

Following elements are represented:
- Background: - to convince the viewer that the display is three dimensional...
- Nodes: - a computer, a network device...
- Traffic: - the amount of traffic flow...
- Protocol: - the type of traffic...
- ...

Node representation

Rendering speed

<table>
<thead>
<tr>
<th>Machine</th>
<th>Rendering speed</th>
<th>Computer (a)</th>
<th>Computer (b)</th>
<th>Computer (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstation</td>
<td>485,000 Δ/sec</td>
<td>0.485 frames/sec</td>
<td>11.5 frames/sec</td>
<td>69 frames/sec</td>
</tr>
<tr>
<td>PC1</td>
<td>30,000 Δ/sec</td>
<td>0.03 frames/sec</td>
<td>0.71 frames/sec</td>
<td>4.3 frames/sec</td>
</tr>
<tr>
<td>PC2</td>
<td>11,000 Δ/sec</td>
<td>0.011 frames/sec</td>
<td>0.26 frames/sec</td>
<td>1.6 frames/sec</td>
</tr>
</tbody>
</table>
Some representation methods allow different *levels of detail*.

In VRML an object may be represented in different ways depending on how large it is.

If the object is near you, it could be represented in detail, but if it is a long way away, the representation could be as simple as a coloured square.

If the distance from the user to the object is smaller than the first range value specified, then the first version is drawn.

If the distance is greater than the last range specified, the last version is drawn.

Draw a line between nodes. A line indicates source and destination, but not the *amount* of traffic:

1. Colour coding (black through red to white for maximum traffic),
2. Line width, and
3. The length of partial lines, as discussed in Eick’s papers.
Trend representation

✔ Graphing
✔ 4D visualization methods
✔ Encode previous on-top-of the current - visual echoes.

Display

Systems

✔ CosmoPlayer VRML viewer,
✔ geomview.

The visualization is not dependant on the navigation or implementation method.
Aggregation

Aggregation Nodes

Implementation #1

✔ A data collector
✔ A web page with... a
✔ Java program loaded as an applet, and a
✔ VRML view of the network.

Web page

```html
<html>
<head>
<title>Sample 3DVNT Page</title>
</head>
<center>
<H1>Sample 3DVNT Page</H1>
</center>
<center>
<embed src="root.wrl" height="600" width="700">
</center>
<center>
<applet code="View1.class" width="100" height="10" mayscript>
<PARAM name="segment" value="MACS">
<PARAM name="port" value="9876">
<PARAM name="host" value="opo.usp.ac.fj">
</applet>
</center>
OK?
</html>
```
Java 3

```java
else {
    if (b.intValue() == -1) {
        val[0] = (float)0.1;
        val[1] = (float)0.1;
        val[2] = (float)0.1;
        appears[a.intValue()].setValue(val);
    } else {
        val[0] = (float)0.0;
        val[1] = (float)1.0;
        val[2] = (float)0.0;
        appears[a.intValue()].setValue(val);
    }
    lastval[a.intValue()] = b.intValue();
}
```

//                System.out.println(line);
}

} catch (IOException e) { System.out.println("Reader: "+ e); }
}

public Browser getBrowser() {
    return browser;
}
```

Summary of topics

In this module, we introduced the following topics:

- Visualization versus computer-graphics
- OpenGL
- (Briefly) Java3D, VTK
- VRML/Java/EAI