CAV Assignment 2

Daniel Holden
s0822954@staffmail.ed.ac.uk
Aim

- Produce visualisation of Human Head volume data
- Attempt to convey “good” or “important” information
  - Skull, Skin, Form & Shape, etc
- Attempt to hide “bad” or “useless” information
  - Artefacts, Noise
- Interactive if possible
Approach

- Ray Traced Volume Renderer
- Step through Volume by some method
- Use Transfer functions to map scalar value to color
- Show skull from different viewpoints
- If skin/bone are clear and visualisation is good then should get mark of 70%

Bonus Marks
- Ray Tracing in 3D
- Trilinear Interpolation
- Volume Illumination
- Shear Warping
Voxel Traversal

- Decide how to traverse into the volume
- Start by showing the volume from different axis
- Decide when to stop traversal
  - Threshold Value
  - Maximum Value
  - Average Value
  - Composite Value
- How you accumulate values as you traverse is important
Transfer Functions

- Map some scalar value to some color/opacity
- Designed to best visualise the data range
- Important to show skin/bone etc
- If unsure design several which show different things well
- It can help to plot histograms to see distributions of density values.
3D Ray Tracing

• Algorithm
  – For each pixel find its clip space coordinates \((x, y)\). These are in the range \((-1, 1)\)
  – Augment with depth value to get near-clip-plane and far-clip-plane coordinates \((x, y, -1)\) and \((x, y, 1)\)
  – Multiply these into world space using some Inverse Projection Matrix and the Inverse View Matrix.
  – These are the endpoints of your viewport pixel ray in world space
Volume Illumination

- Illumination can help us better understand 3D structure of volume information
- Displays Visual Cues such as surfaces
- Highlights important gradients and makes them clear
- Allows for ISO surface display
Volume Illumination

- Voxels can *scatter light*, *reflect light*, or *absorb light*
- Fully modelling scattering is very expensive
- For now just model absorption, reflection
- Use regular Phong Shading Model
- Requires Normals
Estimating Normal from Volume

- 3D Volume has no “normal” only a gradient
- Calculate Gradient using Midpoint Method
- Use it to find vector going in direction required
  - E.G From Most Dense to Least Dense
    
    \[
    nx = V(x-1, y, z) - V(x+1, y, z) \\
    ny = V(x, y-1, z) - V(x, y+1, z) \\
    nz = V(x, y, z-1) - V(x, y, z+1) \\
    \]

    \[ N = (nx, ny, nz) \]
  - Or from current value to required ISO value
Result

- Illuminated Volume
Demo Program

- "include" - Place your header (.h) files here
- "src" - Place your source (.cpp) files here
- "obj" - Intermediate build folder
- "Volume.cpp" - Program entry point
- "head" - Volume data
- "Makefile" - Config file to build the project
- "README" - Instructions for use
Demo Classes

- **Matrix**
  - Contains Matrix classes (for 2x2, 3x3 and 4x4)
  - Feel free to use and extend where needed

- **Vector**
  - Contains Vector classes (for 2, 3, 4)
  - Again, feel free to use and extend

- **Volume**
  - Used to load and access Volume Data
Demo Functions

• Draw()
  - This is called every frame, it is where your logic should go, both for calculations and rendering.
  - Currently it traces in 2D and terminates once it reaches a density over a given value.
  - It then outputs the colour of that density
  - This is the main function you should edit
Demo Functions

• KeyEvent()
  – Here you can add interactive controls for keyboard presses.
  – See also “glutSpecialFunc”, “glutMouseFunc”, “glutMotionFunc” to add other kinds of interaction

• main()
  – You can add initialisation code here
Compiling & Running

• Compiling
  – Providing everything in correct place...
  – Just run “make”
  – Run “make clean” to remove any intermediate build files.

• Running
  – Just run “./Volume”

• Any other problems contact me
How to get started

• Program some form of Transfer Functions
  – Gradient mapping range [0 - 1] to colours
• First trace into the volume in 2D
• Use density data as opacity
• Afterwards attempt to trace in 3D
• Don't attempt to speed up code before it works
Example 2...
Lab session Task

- Designing a really basic transfer function...
FAQ

• Is the head volume meant to be a cube?
  – Yes. If you can account for one axis being smaller great! Otherwise you won’t be marked down.

• Can I use OpenGL to do the 3D viewport?
  – No and it won’t work properly anyway. Always use glVertex just to draw pixels.
  – Please don't use GL_BLEND, do the blending yourself
Any Questions?