Volume Rendering & Classification

Visualisation – Lecture 10
Taku Komura
Institute for Perception, Action & Behaviour
School of Informatics
Volume Rendering ... story so far

- **Volume Rendering**
  - display volume via direct voxel rendering
  - consider transparency in rendering pipeline
    - see through (i.e. into) 3D volumes

- **Image Order Volume Rendering**
  - ray casting through 3D volume
  - intensity transfer function (composite / MIP)
Volume Classification

- **Requirement**: map volume scalar values to opacity, brightness & colour, so that we can
  - see different part with different colour/opacity
    - feature extraction and highlighting
  - similar to colour mapping (lecture 5)
  - transfer function
Opacity Transfer Function

- Mapping scalar value to opacity
- Previously we just used the scalar value as the opacity, but we can do something wiser
  - e.g. isolated bone within volume

Step function.
- looks like a 3D contour surface.

Opacity %

Air

Bone

CT Value

1230

100
Feature Isolation via Opacity

How to visualise the tumour?

- Now we can solve this problem
  - isolate tumour in visualisation using opacity transfer function
  - may also isolate other features to allow relative positioning
Opacity: step transfer function

- Individual features can be isolated at known scalar values
  - suitable scalar value choice dependant on what you want to see

280 units

1200 units
Volume Classification: colour

Gradual transition

Iso-surface like step function.

- cannot show ranges with isosurfaces (e.g. muscle = range of values)
  - isosurfaces (as contours) only show a transition
Colour Volume Classification

Gradual transition

- Thickness shown by amount of colour present in particular region (visualisation of relative density)

Air  Muscle  Bone

Material %

Yellow – Fat
Brown – Tissue
White - Bone
Example: Volume Rendering
Colour/Opacity Feature Highlighting

- many different ways to visualise same volume
  - which is best?
Specifying Transparency

• Allocation of transparency based on scalar value
  - As was done in Lecture 9.
  - not always possible to feature part of interest
  - often interested in edges or changes in scalar value

• Assign transparency based on localised gradient magnitude [Levoy '88]
  - effectively edge detection within volume
Volume Clipping

- Clipping volume in X, Y, Z planes allows “slice-through” visualisation of volumes
  - clip using implicit planes
Example: carotid artery

- Re-examine using volume rendering
Object Order Volume Rendering

- process voxels in order
  - back to front rendering
  - project each to view plane
    - orthographic or perspective projection
    - calculate contribution to target pixel value

- As per object order rendering in CG: consider voxels as objects

- How to visualize the voxels?
  - Just show some blocks for the voxels?
    - Looks like LEGO... Doesn't look nice...
Object Order Voxel Projection

- **Voxel position** determined via projection
  - associated **target pixel** is brightened

- Holes can appear in rendering due to discrete nature of pixels
  - zoom in closer, cell density decreases
  - one voxel projects to one pixel

- **Solution**: spread cell energy across multiple pixels - **splatting**
Splatting

- Project voxel onto image plane splat at a time
- Splat: defined by 3D kernel centred on each voxel
  - projected to 2D footprint on image plane
  - Accumulated back-to-front
  - Size/shape of kernel determines rendering characteristics (sharpness / likelihood of holes)
    - e.g. Gaussian kernel (top)
    - perspective projection: splats non-uniform shape
- One voxel influences many pixels
Splatting

• Kernel size affects image appearance
  – Small sizes generate gaps
  – Large sizes generate blurring
  – Can dynamically change the size for different resolutions

• General rule
  – Size should be larger than or equal to the volume grid spacing
Hybrid rendering: Shear-warp

- shear-warp algorithm
  - hybrid volume rendering approach [Lacroute '94]
  - traverses both image and object space simultaneously in scan-line order
  
  - assumes a orthographic camera model
    - projection perpendicular to image plane
  - Fixed resolution: 1-to-1 correspondence between voxels and pixel
Shearing volume data

- **volume is sheared** so that rays remain perpendicular to base plane
  - lead to **efficient ray computation**
Rotation of camera with *shear-warp*

- Instead of traversing along rays, **visit voxels in a plane**
- regardless of camera viewing angle
Transform to sheared space

- Rays are cast from the base plane voxels at the same place.
- They intersect voxels on subsequent planes in the same location.

• Only one set of interpolation weights needs to be computed for all the voxels in a plane.
• Remember for previous sampling, we needed to recompute the weights for every sample point!
Shear-warp factorisation

• Use **front-to-back ordering**
  – support early termination (last lecture)

• Both **voxels and screen pixels are sampled simultaneously**

• Suitable for SIMD processors (interpolating the data simultaneously for several voxels, summing vector data)
Perspective projection

- We can also handle perspective projection
- Need to shear and **scale**!
Shearing factorisation from 3 different directions

- Doing the shearing for three different views (X, Y, and Z axis)
Final stage of shear-warp

• Final stage of re-sampling (warp) required
  - transform resulting image from sheared space to regular Cartesian space (image plane).
Summary

• **Volume Classification**
  - colour / opacity transfer functions
  - feature extraction and highlighting

• **Object Order Volume Rendering**
  - Volume Rendering via Texture Mapping (H/W)
  - Shear-Warp Algorithm

*Next lecture: volume illumination & vectors*