CMSC 635

Volume Rendering
Volume data

- 3D Scalar Field: $F(x,y,z) = ?$
  - Implicit functions
  - Voxel grid
- Scalar data
  - Density
  - Temperature
  - Wind speed
  - …
Implicit functions

- **Blobs** [Blinn 82]
  - $\exp(-ar^2)$

- **Metaballs** [Nishimura 83]
  - $0 \leq r \leq R_i / 3 : 1 - 3(r / R_i)^2$
  - $R_i / 3 \leq r \leq R_i : 3 (1 - r / R_i)^2 / 2$
  - $r < R_i : 0$

- **Soft Objects** [Wyvill 86]
  - Polynomial approximation for $\exp()$
Voxels

- Sampled volume
  - Usually in a grid
- Measured
  - MRI, CT scan, …
- Computed
  - Sample geometric model
  - Finite element simulation
  - …
Isosurface rendering

- $F(x,y,z) - c = 0$ (for some given $c$)
- Isosurface normal: $\nabla F$
- Implicit: Point repulsion [Witkin 92]
- Voxel: Marching cubes [Lorensen 87]
Marching cubes

- Estimate intersection point on each edge
  - Same criteria (e.g., linear interpolation)
  - Polygons will match

- Use template for polygons
  - $2^8$ possibilities, 15 “unique”
  - Store templates in table
Marching tetrahedra

- Decompose volume into tetrahedra
- Avoids ambiguous “opposite corner” cases
- $2^4 = 16$ cases, 3 unique
  - 0 or 4 points inside (0 triangles)
  - 1 or 3 points inside (1 triangle)
  - 2 points inside (2 triangles)
Dividing cubes

- Find voxels that cross isosurface
- Subdivide to pixel-sized sub-voxels
- Find sub-voxels that cross isosurface
- Plot as shaded points / kernel footprints
Direct volume rendering

- Model as transparent material
  - Color and extinction $C(p)$, $\alpha(p)$
  - Attenuation along ray $p$: $p = r(t)$
    - $\alpha(t) = e^{-\int \alpha(t') dt'}$
  - Attenuated color at $r(t)$
    - $C(t) \alpha(t)$
  - Accumulate attenuated colors along ray
    - $I = \int C(t) \alpha(t) dt$
Simplify volume integral

- **Numeric integration, step size d**
  - \( e^{-\int \Box dt} \approx e^{-\sum \Box d} = \prod e^{-\Box d} = \prod (1-\Box_i) \)
  - Color of ray segment \( i \approx \Box_i C_i \)
  - \( I \approx \sum \Box_i C_i \prod (1-\Box_j) \)

- **Back to front composite**
  - \( C'_i = \Box_i C_i + (1-\Box_i) C'_{i+1} \)
Transfer functions

- Map scalar to color and/or opacity
Appearance

- Additive / pseudo-XRay
- Volume lighting: $N \cdot L, (N \cdot H)^e$
  
  - $N \cdot V = \nabla F \cdot V = \text{Directional derivative}$
  
  - $F_v \approx \left( F(P+kV) - F(P-kV) \right) / 2k$
Rendering methods

- Ray casting
- Splatting
- Texture accumulation
- Shear-warp
Ray casting

- Straightforward numerical integration
- Uniform steps along ray
- Resample volume to sample points
  - Before classification and/or shading
  - After classification and/or shading
Splatting [Westover 90]

- Resample directly onto screen
- Each voxel contributes kernel footprint
- Accumulate back-to-front
Shear-warp [Lacroute 94]
Texture accumulation

- Let texturing hardware resample
- Accumulate back-to-front
- 3D textures
  - Render slices parallel to image plane
  - Shift accesses for $\square F \cdot L$, $\square F \cdot H$
- 2D texture slices
  - Slice sets perpendicular to each axis
  - Choose set most parallel to image plane
Pre-integrated texture [Engel 01]

- Improve approximation for $C_i$ and $\mathbb{T}_i$
  - $\text{Lookup}(\text{start value, end value, d})$
- Dependent lookup
  - 3D texture
    - linear in $d$
  - 2D texture
    - constant $d$
Pre-integrated texture

- a: shading before resampling
- b: shading after resampling
- c: b with interpolated slices
- d: pre-integrated, same slice set as b