1 PhysBAM

1.1 Installing Geometry

Provide a URL, show how to start the build process.

1.2 Rendering Level Sets

- Show how to view a level set in the viewer.
- Show how to render a level set.
- Also show the relevant bits of the scene file.

2 Implicit Surfaces

2.1 Simple Properties

- Implicit function $\phi : \mathbb{R}^3 \to R$
  - Inside: $\phi < 0$
  - Outside: $\phi > 0$
  - Interface: $\phi = 0$
- Need not be a level set at this stage
- Efficient inside/outside tests
- Recall that $\nabla\phi$ is orthogonal to the level sets of a function
- $\mathbf{N} = \frac{\nabla\phi}{\|\nabla\phi\|}$

2.2 Ray Tracing Implicit Functions

- Generic representation
  - Ray: $\mathbf{x}(t) = \mathbf{r} + t\mathbf{u}$
  - Implicit function along ray: $f(t) = \phi(\mathbf{x}(t)) = \phi(\mathbf{r} + t\mathbf{u})$
  - Find first solution to $f(t) = 0$ with $t > 0$
  - Surface normal is available for surface shaders: $\mathbf{N} = \frac{\nabla\phi}{\|\nabla\phi\|}$
- Regular grid representation
  - Assume ϕ at nodes for simplicity
  - Defined inside cells by interpolation of corner values
  - Assume linear interpolation. Write this out in 2D.
  - Walk the grid cell by cell (Bresenham's line algorithm - don't go into detail)
  - Check for interface in that cell
    - Prune if possible
    - Find root of polynomial (cubic in 3D)
  - Compute normal $\mathbf{N} = \frac{\nabla\phi}{\|\nabla\phi\|}$
    - Compute $\nabla\phi$ at nodes with central differenes
    - Interpolate node gradients
    - Equivalent to central differences on interpolatead values

3 Level Sets

3.1 Definition

- Special type of implicit surface representation
- Signed distance function
  - Inside: $\phi < 0$
  - Outside: $\phi > 0$
  - Interface: $\phi = 0$
- Draw some example contours for an object
- Mostly smooth, but will have some sharp features

3.2 Level Set Property

- Choose point off interface, draw circle tangent to interface
- Level Set Property: $\|\nabla\phi\| = 1$ (Motivate this)
- $\mathbf{N} = \frac{\nabla\phi}{\|\nabla\phi\|} = \nabla\phi$

3.3 Curvature

- Surface curve $\gamma(s) = \begin{pmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\end{pmatrix}$ is a smooth curve along surface through point $\gamma(0) = \mathbf{x}$
- $\phi(\gamma(s)) = 0$
- Normal
  - $0 = \frac{d}{ds}\phi(\gamma(s)) = \frac{\partial\phi}{\partial x}\frac{d\gamma_1}{ds} + \frac{\partial\phi}{\partial y}\frac{d\gamma_2}{ds} + \frac{\partial\phi}{\partial z}\frac{d\gamma_3}{ds} = \nabla\phi\cdot\gamma'$
  - $\mathbf{T} = \gamma'$
  - $\nabla\phi$ is orthogonal to every curve on surface, normal direction
  - Points to side of interface with $\phi > 0$
- Curvature
  - $0 = \frac{d^2}{ds^2}\phi(\gamma(s)) = \frac{d}{ds}(\nabla\phi\cdot\gamma') = \gamma'\cdot\frac{d}{ds}\nabla\phi + \nabla\phi\cdot\gamma'' = \gamma'\cdot(\mathbf{H}\gamma') + \nabla\phi\cdot\gamma''$
  - Hessian: $\mathbf{H}$
  - $\nabla\phi = \mathbf{N}$, where normal points outwards
  - κ is signed curvature, positive for sphere
  - $\gamma'' = -\kappa\mathbf{N}$ (Note sign of κ and direction of $\mathbf{N}$)
  - $\mathbf{T}^T\mathbf{H}\mathbf{T} = \kappa$
  - $0 = \frac{d}{ds}(\nabla\phi\cdot\nabla\phi) = 2\mathbf{H}\mathbf{N}$
  - $\mathbf{N}^T\mathbf{H}\mathbf{N} = 0$
  - Principle curvatures can be deduced from $\mathbf{H}$.
  - 2D
    - Eigenvalues: 0, κ

    * Eigenvectors: $\mathbf{N}$, $\mathbf{T}$
    * $\nabla^2\phi = \mathrm{tr}(\mathbf{H}) = \sum_i \lambda_i = \kappa$
- 3D
    * Eigenvalues: 0, κ_1, κ_2
    * Eigenvectors: $\mathbf{N}$, corresponding principle curvature directions
    * $\nabla^2\phi = \mathrm{tr}(\mathbf{H}) = \sum_i \lambda_i = \kappa_1 + \kappa_2 = 2\kappa_{mean}$