Designing with Distance Fields
From Concept Modeling to Detailed Carving

Sarah F. Friskin, Tufts University
Ronald N. Perry, Mitsubishi Electric Research Labs

Distance Fields

- An object’s distance field represents, for any point in space, the distance from that point to the object
  - The distance can be signed to distinguish between the inside and outside of the object
Distance Fields

- Distance fields are *implicit representations* of shape ...
  - See *Introduction to Implicit Surfaces* (J. Bloomenthal, ed.), 1997

- The surface of the shape is an *implicit surface*
  - An implicit surface is an iso-contour of an implicitly defined scalar function

---

Distance Fields

- An example of an implicit representation, $F(x)$, where $x \in \mathbb{R}^3$

  ![A 2D cross section of $F(x)$](image1)
  ![An iso-contour of $F(x)$ where $F(x) = 0.28$](image2)
Distance Fields

- Typically, for a shape represented by a distance field, the shape’s boundary, $\Omega$, is the zero-valued iso-surface of the distance function.
  - i.e., for an implicitly defined distance function, $dist(x)$, $\Omega$ is the set of all points where $dist(x) = 0$

Distance Fields

- General distance function
  
  $$dist(x) = \text{Norm}(x - S(x))$$
  
  where $\text{Norm}(u)$ is a metric that decreases monotonically with $||u||$, and $S(x)$ is a point on the boundary $\Omega$.

- Minimum distance
  
  $S(x) = s^*$, where $s^*$ is on $\Omega$ and
  
  $$|\text{Norm}(x - s^*)| \leq |\text{Norm}(x - S)|, \forall S \in \Omega$$

- Euclidean norm
  
  - Minimum Euclidean distance
    
    $$dist(x) = \pm \sqrt{(x - s^*)^2 + (y - s^*)^2 + (z - s^*)^2}, \text{ for } x = (x, y, z)$$
    
    = the signed magnitude of the vector from $x$ to $s^*$.
**General Properties**

- Distance fields are defined everywhere in space
  - Not just on the surface like boundary representations
- It is trivial to determine whether a point is inside, outside or on the boundary of a shape
  - Evaluate \( F(x) \) and compare it to the value of the iso-surface
- Gradients of the distance field provide geometric information
  - On the surface
    - The gradient is normal to the surface
  - Off the surface
    - The gradient points in the direction of the closest surface point

**Continuity Properties**

- Distance fields are \( C^0 \) continuous everywhere
- Euclidean distance fields are \( C^1 \) continuous except at boundaries of Voronoi Regions
  - \( C^1 \) near smooth sections of the boundary
    - Distance field is linear near linear sections of the boundary
  - Not \( C^1 \) near corners or the medial axis
Operations on Distance Fields

- **Constructive Solid Geometry** or CSG is the result of a Boolean operations applied to primitive, aggregate, or CSG objects.
  - Boolean operations on distance fields are fast and simple.

### Boolean Operations

- **Union**: \( A \cup B \)
  
  \[ \text{dist}(A \cup B) = \max(\text{dist}(A), \text{dist}(B)) \]

- **Intersection**: \( A \cap B \)
  
  \[ \text{dist}(A \cap B) = \min(\text{dist}(A), \text{dist}(B)) \]

- **Difference**: \( A - B \)
  
  \[ \text{dist}(A - B) = \min(\text{dist}(A), -\text{dist}(B)) \]
Operations on Distance Fields

• Boolean operations are fast and simple

Union of two shapes
Union of two distance fields

Advantages

• Conceptual advantages over outlines
  − Distance fields represent more than just the object outline
    • Represent the object interior, exterior, and its boundary (useful for CSG operations and physical simulation)

Shape's distance field
Advantages

- Conceptual advantages over outlines
  - Distance fields represent more than just the object outline
    - Represents an infinite number of offset surfaces

Boundary offsets

Advantages

- Conceptual advantages over outlines
  - Gains in efficiency and quality because distance fields vary smoothly ($C^0$ continuous) and throughout space

Intensity profile of the standard representation is discontinuous at boundaries

Intensity profile of the distance field is $C^0$ continuous throughout space
Advantages

- Conceptual advantages over outlines
  - Gradient of the distance field yields
    - Surface normal for points on the edge
    - Direction to the closest edge point for points off the edge

History – Rendering and Processing

- Distance fields are a specific example of implicit functions (see "Introduction to Implicit Surfaces", ed. Bloomenthal, 1997)

- Rendering and processing
  - Tessellation
    - Szeliski and Tonneen, "Surface modeling with oriented particle systems", SIGGRAPH, 1992
    - Heckbert and Witkin, "Using particles to sample and control implicit surfaces", SIGGRAPH, 1994
  - Ray tracing
History – Applications

• CAD
  – Offsetting
  – Tolerancing
  – Rounds and files
  – Swept surfaces and volumes
  – Simulation
    – Biswas and Shapiro, "Approximate distance fields with non-vanishing gradients", Graphical Models, 2004

• Image processing
  – Image segmentation
    – Yang, Staib, and Duncan, "Neighbor-Constrained Segmentation with Level Set Based 3D Deformable Models," IEEE Trans Med Imaging, 2004
  – e.g., watershed segmentation
  – Shape matching
    – Lavalle and Szeliski, "Recovering the Position and Orientation of Free Form Objects from Image Contours Using 3D Distance Maps", IEEE PAMI, 1995

Original image, watershed segmentation, and segmented cells: from www.cellprofiler.org
History – Applications

- **Medical**
  - 3D surface reconstruction from cross sections
  - Navigation during virtual surgery
  - Medial axis representation for animation of the Virtual Human
    - Gagvani and Silver, Animating volumetric models, Graphical Models, 2001

  Animating the Virtual Human: from Gagvani and Silver, 2001

- **Robotics**
  - Path planning
  - Collision detection
    - Fisher and Lin, "Deformed Distance Fields for Simulation of Non-Penetrating Flexible Bodies", Eurographics Wkshop on Comp Anim and Modeling, 2001
  - Haptics
History – Applications

• Simulation
  – Modeling continually varying heterogeneous materials
  – Level sets
    – Osher and Sethian, “Fronts Propagating with Curvature-Dependent Speed, Algorithms Based on Hamilton-Jacobian Formulation”, J. Computational Physics, 1988

Fedkiw, Stam, and Jensen

Neuen, Fedkiw, and Jensen

Enright, Marschner, and Fedkiw

History – Applications

• Reconstructing 3D models from range data
  – Unorganized points, projected distances, 3D distance fields
    – Hoppe et al., “Surface Reconstruction from Unorganized Points”, SIGGRAPH, 1992
    – Bajaj, Bernardini, and Xu, “Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans”, SIGGRAPH 1995
    – Carr et al., “Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001
    – Hilton et al., “Reliable Surface Reconstruction from Multiple Range Images”, European Conf. on Computer Vision, 1996
    – Sagaawa, Nishino, and Ikeuchi, “Robust and Adaptive Integration of Multiple Range Images with Photometric Attributes”, 2001
    – Frisken and Perry, “Efficient Estimation of 3D Euclidean Distance Fields from 2D Range Images”, Symposium on Volume Visualization, 2002
History – Applications

- Distance Fields for Design

- Distance Fields for Design
  - Gaylean and Hughes, "Sculpting: an Interactive Volume Modeling Technique", SIGGRAPH 1991
History – Applications

• Distance Fields for Design
  • Avila and Sobierajski, "A Haptic Interaction Method for Volume Visualization, IEEE Visualization 1996

![Tree Image]

History – Applications

• Distance Fields for Design

![Chair and Guitar Image]
**History – Applications**

- **Distance Fields for Design**

- **Distance Fields for Design**
  - Cani (Gascuel), "An Implicit Formulation for Precise Contact Modeling Between Flexible Solids", SIGGRAPH 1993
History – Applications

• Distance Fields for Design
  • Desbrun and Cani, "Active Implicit Surface for Animation", Graphics Interface, 1998

• Distance Fields for Design
  • Blanch, Ferley, Cani, Gascuel "Non-Realistic Haptic Feedback for Virtual Sculpture", Rapport de recherche RR-5090, 2004
History – Applications

- Distance Fields for Design
  - Perry, Frisken, “Kizamu: a system for sculpting digital characters”, SIGGRAPH 2001
**History – Applications**

- **Distance Fields for Design**

**Representing Distance Fields**

- **Implicit representation**
  - e.g., sphere, \( \text{dist}(x, y, z) = R - \sqrt{(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2} \)

- Distances are computed at query points as needed for rendering or processing
- Complex models can be represented via CSG
  - Precise but slow for complex models
Representing Distance Fields

- **Sampled volumes**
  - Distances are computed and stored in a regular 3D grid
  - Distances at non-grid locations are interpolated

- A 2D shape and three signed distance values
- A regularly sampling of the distance field
- A 2D image depicting the distance field

---

Representing Distance Fields

- **Sampled volumes**
  - Smooth surfaces are well represented by a relatively small number of samples
    - Frisken (Gibson), "Using Distance Maps for Smooth Surface Representation in Sampled Volumes", IEEE Symposium on Vol Vis, 1998

- Radius = 30 voxels
  - 100 x 100 x 100
- Radius = 3 voxels
  - 10 x 10 x 10
- Radius = 2 voxels
  - 10 x 10 x 10
- Radius = 1.5 voxels
  - 10 x 10 x 10
Representing Distance Fields

- **Sampled volumes**
  - For detailed models, the distance field must be sampled at high enough rates to avoid aliasing during reconstruction and rendering.
  - Regularly sampled volumes have
    - Slow processing times
    - Large memory requirements
    - Limited resolution

Representing Distance Fields

- **Addressing memory requirements and processing speed**
  - Speed up distance computation using approximating distance transforms.

```
3 2 2 3 4 4
2 1 1 2 2 3
1 0 0 1 1 2
2 1 0 0 0 1
3 2 1 0 0 1
4 3 2 1 1 2
```

**Binary 2D shape**  **Chessboard distance**
Representing Distance Fields

- Addressing memory requirements and processing speed
  - Speed up distance computation using hardware

- E.g., points

2D distance field of a point

A 3D cone with its apex at the point has z-coordinates that correspond to the point’s distance field

- e.g., lines

2D distance field of a line

A 3D folded plate with its apex along the line has z-coordinates that correspond to the line’s distance field
Representing Distance Fields

- Addressing memory requirements and processing speed
  - Fast distance computation using hardware

![Hardware-generated Voronoi diagram of a chevron shape](image1)

![Projected 3D geometry used to approximate the 2D distance field of the chevron](image2)

Using hardware generated distance fields to control particle dynamics
Representing Distance Fields

- Addressing memory requirements and processing speed
  - Reducing the number of distance samples
    - Distance shells (compute accurate distances in a narrow region near the boundary)
    - Level sets for propagating accurate distances throughout the volume
      - Breen, Mauch, and Whitaker, "3D Scan Conversion of CSG Models into Distance Volumes", IEEE Symposium on Vis Vis, 1998
  - Boundary-limited Octrees
    - Jones and Satherly, "Shape Representation using Space Filled Sub-Voxel Distance Fields", IEEE International Conf. on Shape Modeling and Apps, 2001
    - Strain, "Fast Tree-based Redistancing for Level Set Computations", J. Computational Physics, 1999
  - ADFs

Boundary-limited quadtree
Representing Distance Fields

- Addressing memory requirements and processing speed
  - Reducing the number of distance samples
    - Boundary-limited octrees and quadtrees
      - Jones and Satherly, "Shape Representation using Space Filled Sub-Voxel Distance Fields", IEEE International Conf. on Shape Modeling and Apps, 2001
      - Strain, "Fast Tree-based Redistancing for Level Set Computations", J. Computational Physics, 1999
    - ADFs

Adaptively Sampled Distance Fields

- Detail-directed sampling of the distance field

Sample at low rates where the distance field is smooth. Sample at higher rates only where necessary (e.g., near corners).
Adaptively Sampled Distance Fields

- Detail-directed sampling of the distance field
  - High sampling rates only where needed
    - Fewer distance samples to compute
    - Less memory required
    - Faster to process

- Spatial data structure
  - Fast localization for efficient processing

- ADFs are defined very generally: they consist of
  - Adaptively sampled distance values ...
  - Organized in a spatial data structure ...
  - With a method for reconstructing the distance field from the sampled distance values

Various Instantiations of ADFs

Example of a quadtree-based 2D ADF
Various Instantiations of ADFs

Example of a wavelet-based 2D ADF

Example of a multi-resolution triangulation-based 2D ADF
**Octree-based ADFs**

- A set of distance values is stored for each leaf cell in the octree
- Distances and gradients between the sample points are reconstructed by interpolating the stored values

**Spatial structure of an octree**

**Tree structure of an octree**

**Locally Implicit Representation**

- Reconstruction from samples
  - Distances at positions between sample points are reconstructed from sampled distances
    - The sampling breaks the distance field into a set of spatially-limited local implicit representations

**Sampling on a rectilinear grid with a bi-linear reconstruction function breaks the distance field into locally implicit fields**
Adaptively Sampled Distance Fields

- Rendering
  - There are three basic approaches
    - Ray tracing
    - Tessellation
    - Point models

Ray Casting ADFs

- For each pixel
  - Cast ray(s) into the ADF octree
  - Locate intersection of ray with the object surface
  - Shade the image pixel according to the gradient of the distance field at the intersection point

Cast each ray into the ADF Octree
Determine ray-surface intersections analytically assuming a bilinear surface
Ray Casting ADFs

- To improve rendering times
  - Render from coarse to fine resolution
  - Local redraw during editing
  - Use adaptive ray casting
    - Subdivide the image into a hierarchy of tiles according to perception-based predicates
    - Pixels in tiles greater than 1x1 are bilinearly interpolated
    - Achieve 6x speedup (10x when supersampling)

Ray Casting ADFs

- Adaptive ray casting

Rendered via adaptively ray casting

Rays cast to render part of the image on the left
Point-based Rendering of ADFs

- Generate points on the surface and render the points
  - Seed each boundary leaf cell with randomly placed points, number of points proportional to cell size
  - Relax the points onto the ADF surface using the distance field and gradient
  - Optionally shade each point using the distance gradient
  - Can generate 800,000 Phong-shaded points in 0.2 seconds

Original points seeded in boundary leaf cells
Points after relaxation onto the surface

Point-based Rendering of ADFs

An ADF rendered as points at two different scales
Tessellating ADFs

- Convert ADFs to triangle models and render interactively with hardware
- Use the SurfaceNets triangulation algorithm to create triangle models on-the-fly
    - Topologically consistent
    - Good-quality triangles
    - Fast – 200,000 triangles in 0.37 seconds (in 2001)
    - LOD models

Tessellating ADFs

- LOD triangle models
  - The octree hierarchy provides a natural structure for creating level-of-detail triangle models
Tessellating ADFs

- View-dependent triangle generation
  - Pope, Frisken and Perry, "Dynamic Meshing Using Adaptively Sampled Distance Fields", SIGGRAPH Sketch, 2001

Very large data models can be tessellated dynamically according to viewing parameters.

ADF Generation

- ADF Generation
  - Requires a means for computing the distance to the shape at arbitrary points
  - Top-down generation
    - Compute distances to the shape from each corner of the root cell
    - Compute a representation error for the cell
    - Recursively subdivide cells if they contain a boundary and their representation error is larger than a specified threshold

Initialize root cell

Recursively subdivide
**ADF Editing**

- **Editing is localized generation**
  - The ADF is regenerated inside cells that are overlapped by the tool's bounding volume
  - Regeneration blends the distance fields of the ADF and the tool using CSG operations

During editing, the ADF within a region near the tool's bounding box is regenerated using the appropriate CSG operation to blend the distance fields of the tool and the object.

---

**Detailed Carving**

- **Kizamu**

- **Provides**
  - Generation of parts
    - From stock distance functions (spheres, cubes, cones, cylinders, offsets of 3D Bezier curves)
    - Using CSG of stock parts
    - Via a scripting interface
    - From height fields and range data
    - Extrusion and revolution of 2D ADFs
  - Detailed carving using a variety of brushes and brush paradigms
    - Surface following
    - Pressure sensitive pen input
    - Lathing tools
  - Various rendering methods
    - Ray tracing (surface and volume)
    - Point-based rendering
    - Tessellation

- **History**
- **Infinite undo and redo**
Detailed Carving

- Surface of revolution
- Extruded shape

- Freeform model carved from a sphere
- Detailed model of stones from range data
Kizamu Demo

Concept Modeling

- Input from various sources
  - ADFs can be generated from existing 3D models
    - E.g., from triangle models or Bezier patches
  - ADFs can be generated from range data
    - Data from 3D scanners, pseudo-depth cameras
    - Elevation maps
    - Synthetic range data (e.g., from a texture generator)
- Preliminary construction in Kizamu
  - Stock distance functions can be combined via CSG
    - Spheres, rectangular solids, cylinders, cones, etc.
- Need a way to generate concept models that can be imported into Kizamu
  - Fast
  - Intuitive
  - Expressive
Concept Modeling

- **Creature Feature**
  - A prototype system for creating organic concept models
    - Shapes are based on a curved skeleton with cross-sections along the skeleton defined by 2D curves
    - Interface for creating the skeleton and profiles were inspired by previous work
      - Grimm, "Implicit Generalized Cylinders using Profile Curves", Implicit Surfaces, 1999

Skeleton sketching (Grimm, 1999)
Cross-sectional profiles (Grimm, 1999)

Creature Feature

Add or delete points on the skeleton curve
Drag control points on the curve or on its shadows to edit the shape of the curve
Curve overdrawing is provided
**Creature Feature**

Multiple curves are supported

The current prototype uses circular profiles

The lofted creature

**Creature Feature**

Sketch Skeleton
Creature Feature

Next steps
- Use more complex profiles to define cross sections along the skeleton
- Provide an interface for sketching and editing the 2D profiles as 2D ADFs and/or curves

Enhancements

- Accurate representations for analysis and re-tessellation
  - Feature sensitive surface extraction from volume data, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p.57-66, August 2001
  - Ju et al., "Dual Contouring of Hermite Data", SIGGRAPH 2002
  - Huang et al., "A Complete Distance Field Representation", IEEE Visualization 2001

- Coloring and texturing
  - DeBry et al., "Painting and rendering textures on unparameterized models", SIGGRAPH, 2002
  - Benson and Davis, "Octree textures", SIGGRAPH, 2002
Enhancements

• Improved representation to reduce memory requirements and improve processing speeds
  - Biquadratic ADF cells for more compact representation of curved surfaces and edges
    - US patent 6,396,492, "Detail-directed Hierarchical Distance Fields"
  - Special cells for thin structures, exact corners and edges
    - US patent application "Method for Generating an Adaptively Sampled Distance Field of an Object with Specialized Cells"

Bilinear cell
- 4 distance samples
- Bilinear interpolation
- Represents distance to lines exactly

Biquadratic cell
- 9 distance samples
- Biquadratic interpolation
- Represents distance to curves more accurately
Next Steps

- There are many directions to go
  - Complete Creature Feature and integrate it into Kizamu
  - Take Kizamu beyond the research prototype stage
    - Add more brushes (e.g., smoothing) and features
    - Design and build a reasonable user interface
    - Add coloring and texturing
    - Add high quality rendering
  - Explore deformation and modeling other physical phenomena using ADFs
  - Explore applications in CAD, entertainment, medicine, ...
- ...

Many Thanks

- Ronald Perry
- Elena Jakubiack
- SMI conference organizers