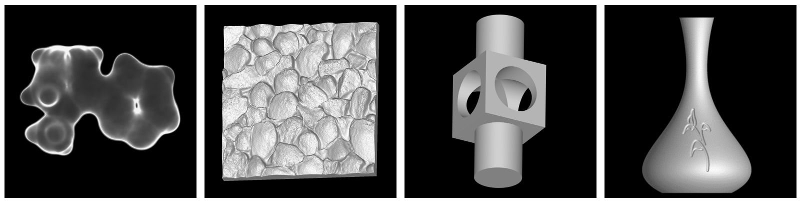
Adaptively Sampled Distance Fields (ADFs)

Representing Shape for Computer Graphics



Sarah F. Frisken and Ronald N. Perry
Mitsubishi Electric Research Laboratories
2-28-2002

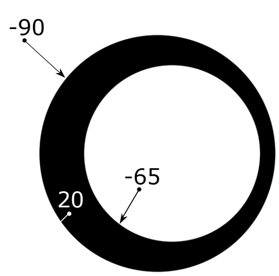
Outline

- Introduction to ADFs
  - definition, advantages, instantiations, algorithms
- Accuracy and Benchmarks
- Technology status
- Demonstration
- Business Opportunities

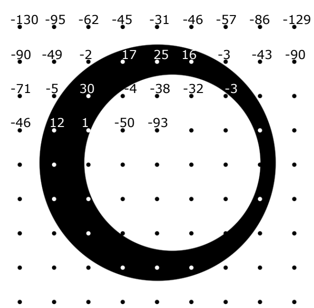
Distance Fields

- A distance field is a scalar field that
  - specifies the distance to the surface of a shape ...
  - where the distance may be signed to distinguish between the inside and outside of the shape
- Distance
  - can be defined very generally (e.g., non-Euclidean)
  - minimum Euclidean distance is used for most of this presentation (with the exception of the volumetric molecules)

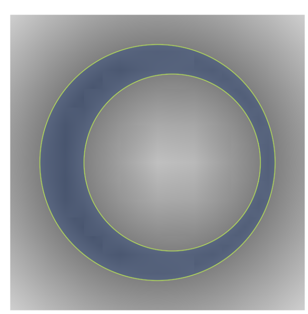
Distance Fields



2D shape with sampled distances to the surface

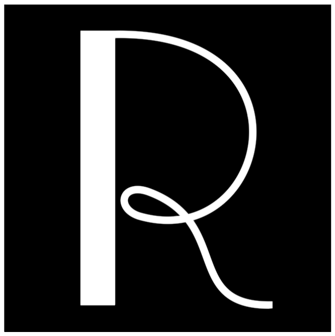


Regularly sampled distance values

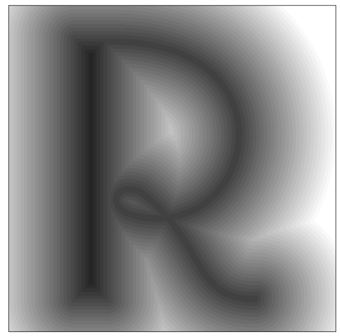


2D distance field

2D Distance Field

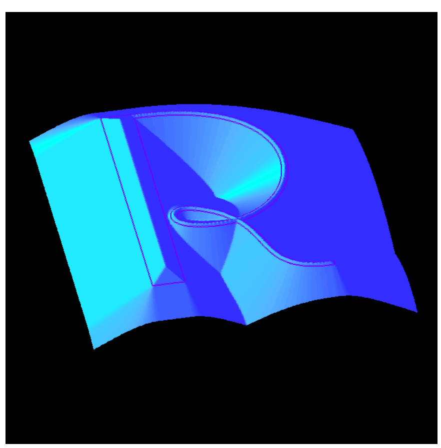


R shape



Distance field of R

2D Distance Field



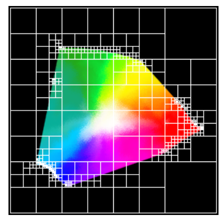
3D visualization of distance field of R

Shape

- By *shape* we mean more than just the 3D geometry of physical objects. Shape can have arbitrary dimension and be derived from simulated or measured data.

Color printer

⇒



Color gamut

Conceptual Advantages of Distance Fields

- Represent more than the surface
  - object interior and the space in which the object sits
- Gains in efficiency and quality because
  - distance fields vary "smoothly"
  - are defined throughout space
- Gradient of the distance field yields
  - surface normal for points on the surface
  - direction to closest surface point for points off the surface

Practical Advantages of Distance Fields

- Smooth surface reconstruction
  - continuous reconstruction of a smooth field
- Trivial inside/outside and proximity testing
  - using sign and magnitude of the distance field
- Fast and simple Boolean operations
  - intersection: dist(A∩B) = min(dist(A), dist(B))
  - union: dist(A∪B) = max(dist(A), dist(B))
- Fast and simple surface offsetting
  - offset by *d*: dist(A_{offset}) = dist(A) + *d*
- Enables geometric queries such as closest point
  - using gradient and magnitude of the distance field

Sampled Distance Fields

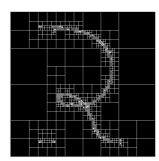
- Similar to sampled images, insufficient sampling of distance fields results in aliasing
- Because fine detail requires dense sampling, excessive memory is required with *regularly* sampled distance fields when *any* fine detail is present

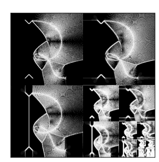
Adaptively Sampled Distance Fields

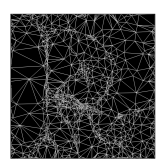
- Detail-directed sampling of a distance field
  - High sampling rates only where needed
- Spatial data structure (e.g., an octree)
  - Fast localization for efficient processing
- Reconstruction method (e.g., trilinear interpolation)
  - For reconstructing the distance field and gradient from sampled distance values

ADF Instantiations

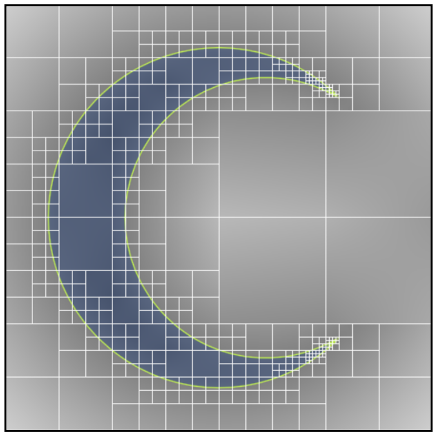
- Spatial data structures
  - octrees
  - wavelets
  - multi-resolution tetrahedral meshes …
- Reconstruction functions
  - trilinear interpolation
  - B-spline wavelet synthesis
  - barycentric interpolation ...







ADFs - A Comprehensive Representation



ADFs provide:

- spatial hierarchy
- distance field
- object surface
- object interior
- object exterior
- surface normal (gradient at surface)
- direction to closest surface point (gradient off surface)

ADFs consolidate the data needed to represent complex objects

ADFs - A Unifying Representation

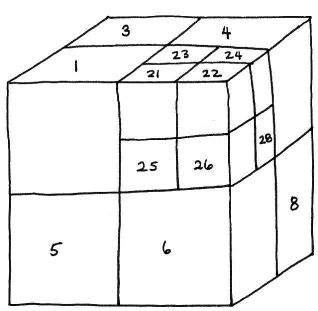
- Represent surfaces, volumes, and implicit functions
- Represent sharp edges, organic surfaces, thin-membranes, and semi-transparent substances
- Consolidate multiple structures for complex objects (e.g., for collision detection, LOD construction, and dynamic meshing)
- Can store auxiliary data in cells or at cell vertices (e.g., color and texture)

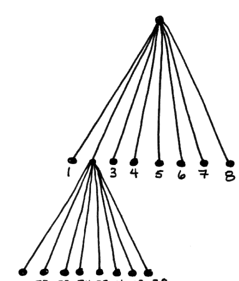
Algorithms for Octree-based ADFs

- Specifics of octree-based ADFs
- Generating ADFs
- Editing ADFs
- Rendering ADFs
- Generating point models from ADFs
- Triangulating ADFs
- Surfacing ADFs
- Hierarchical transmission of ADFs

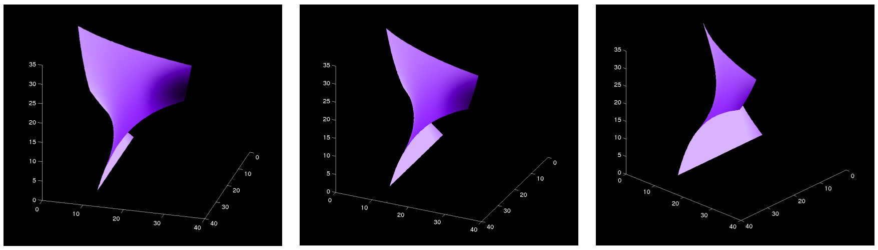
Octree-based ADFs

- A distance value is stored for each cell corner in the octree
- Distances and gradients are estimated from the stored values using trilinear reconstruction



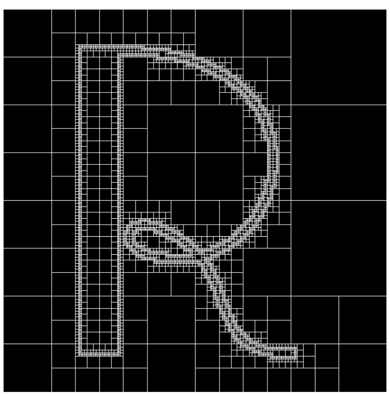


Reconstruction

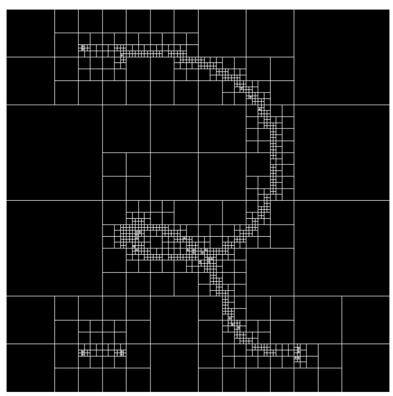


A single trilinear field can represent highly curved surfaces

Comparison of 3-color Quadtrees and ADFs



87,881 cells (3-color)

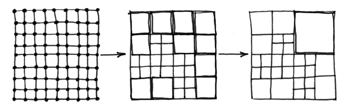


1473 cells* (ADF)

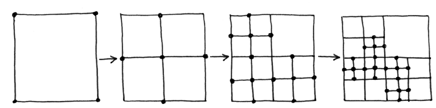
*new research: high order interpolants significantly reduce cell count

Generation

- Bottom-up generation
  - Fully populate
  - Recursively coalesce



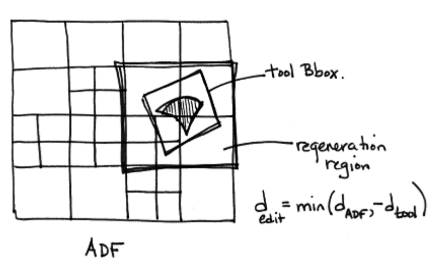
- Top-down generation
  - Initialize root cell
  - Recursively subdivide



- Tiled Generation
  - Top-down generation within localized tiles
  - Reduced memory requirements, better memory coherency, reduced computation

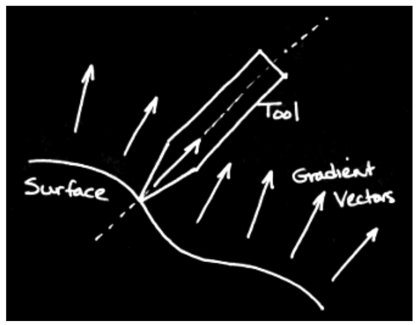
Editing

- Editing is a localized re-generation
  - determine minimum overlap region between tool and the object ADF
  - perform Boolean operation (e.g., subtraction) on the distance fields of the tool and the object in the overlap region



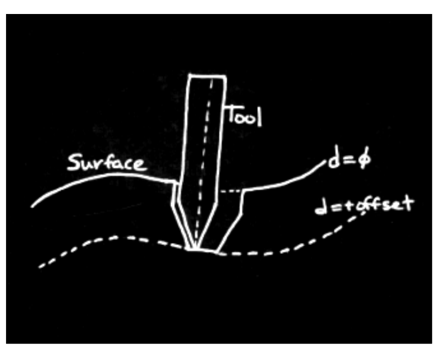
Editing – Sculpting Interface

- Surface following
- Distance-based constraints
- Control-point editing



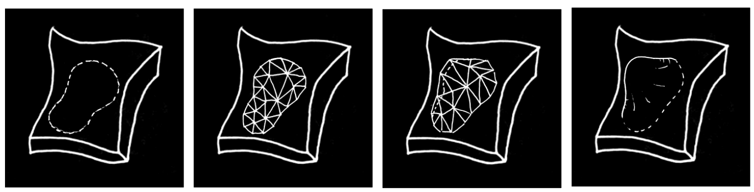
Editing – Sculpting Interface

- Surface following
- Distance-based constraints
- Control-point editing



Editing – Sculpting Interface

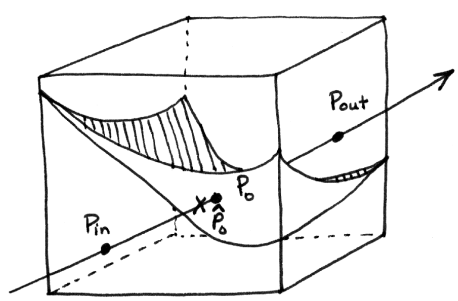
- Surface following
- Distance-based constraints
- Control-point editing



Rendering via Ray Casting

Ray-surface Intersection with a Linear Solver

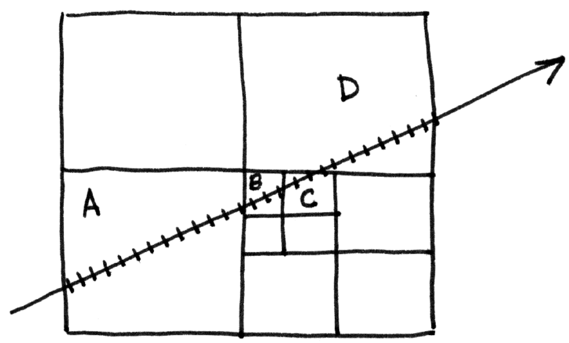
- Assume that distances vary linearly along the ray
- Determine the zero-crossing within the cell given distances at the points where the ray enters and exits the cell



$$\hat{P}_0 = P_{in}\cdot(1-t) + P_{out}\cdot t$$

Ray Casting
Volume Rendering

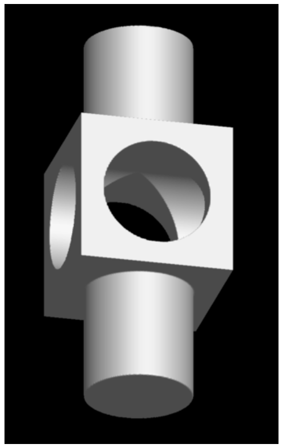
- Colors and opacities are accumulated at equally spaced samples along each ray
- Use octree and distance field to accelerate volume rendering



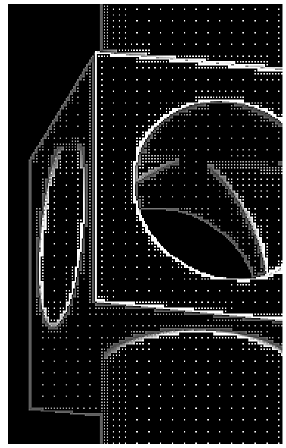
Adaptive Asynchronous Ray Casting

- Adaptive rendering
  - the image region to be rendered is divided into a hierarchy of image tiles
  - the subdivision of each tile is guided by a perceptually-based predicate
  - pixels within image tiles of size greater than 1x1 are bilinearly interpolated to produce the image
  - rays are cast into the ADF at tile corners and intersected with the surface using the linear solver
- Processing occurs
  - asynchronously
  - upon user request
  - to update edited regions

Adaptive Asynchronous Ray Casting

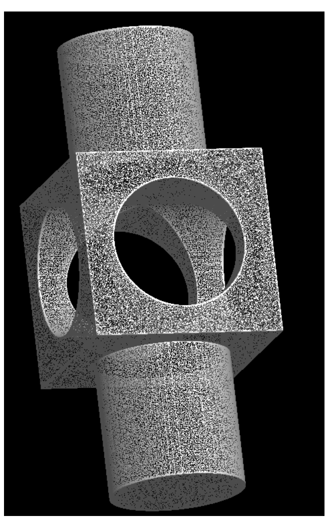


Adaptively ray cast ADF



Rays cast to render part of the left image

Generating Point Models from ADFs



- Points are randomly seeded in boundary leaf cells and moved to the surface
- Fast
  - 1,100,000 points in 0.12s (Pentium IV)
- Can be detail-directed
  - points can be evenly distributed or concentrated near surface detail

Triangulating ADFs

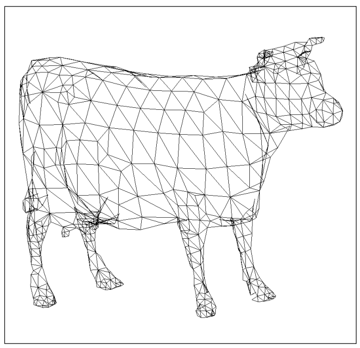
- ADFs can be triangulated using a fast new triangulation method
- Triangulation is efficient
  - 300,000 triangles in 0.37 seconds, Pentium IV
  - 3,000 triangles in < 0.01 seconds
- The triangulation produces models that are orientable and closed

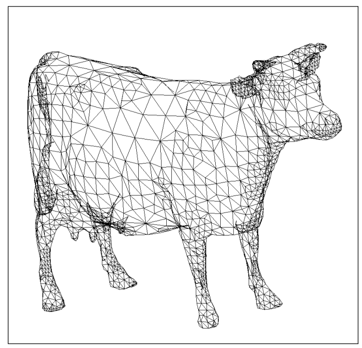
Triangulation Algorithm

- Seed
  - Assign a vertex to each boundary leaf cell of the ADF, initially placing vertices at cell centers
- Join
  - Join vertices of neighboring cells to form triangles
- Relax
  - Move vertices to the surface using the distance field
- Improve
  - Move vertices over the surface towards their average neighbors' position to improve triangle quality

Triangulation – Level-of-Detail

- The octree is traversed and vertices are ***seeded*** into boundary cells whose maximum error satisfies a user-specified threshold
- Cells below these cells in the hierarchy are ignored



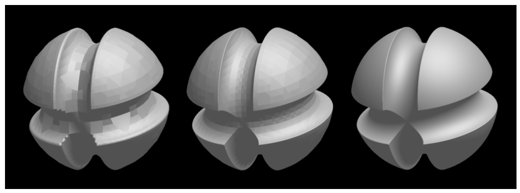


Surfacing ADFs

- Off-the-shelf solution
  - generate a dense point model or a detail-directed triangle model
  - use Geomagic Studio 4 to create NURBS
- ADF-specific approach
  - exploit detail-directed sampling to identify initial patches
  - refine patches using an optimization approach
  - use the distance field to compute surface error and guide refinement

Hierarchical Transmission of ADFs

- ADF hierarchy allows progressive transmission



- ADF hierarchy allows transmission of sub-volumes for localized processing

Accuracy and Benchmarks

- Surface accuracy summary
- Timing

Surface Accuracy Summary

- Planar surfaces
  - can be reconstructed to floating point precision from a small number of sample points
- Curved surfaces
  - limited by the maximum cell error (an ADF generation parameter)
  - level 7 ADF achieves 30 micron accuracy for a 1 meter diameter sphere
- Edges and corners
  - limited by maximum ADF level (an ADF generation parameter)
  - level 13 ADF achieves 10 micron accuracy for a $(8\ cm)^3$ part

Timing

- Generation and Editing
  - Approximately 300,000 cells per second (Pentium IV)
  - 1 meter sphere (at 31 micron accuracy) in 0.265 seconds
  - 1 meter box (at 85 micron accuracy) in 0.310 seconds
- Rendering
  - Asynchronous, adaptive, on-demand ray casting provides interactive rendering
- Point generation
  - 9.2 million points per second (Pentium IV)
- Triangle generation
  - 800,000 triangles per second (Pentium IV)

Technology Status

- Research papers
- Patents
- ADF library

Research Papers

- "Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics", SIGGRAPH 2000 Conference Proceedings
- "Kizamu: A System For Sculpting Digital Characters", SIGGRAPH 2001 Conference Proceedings
- "Computing 3D Geometry Directly from Range Images", SIGGRAPH 2001 Conference Abstracts and Applications
- "A Computationally Efficient Framework for Modeling Soft Body Impact", SIGGRAPH 2001 Conference Abstracts and Applications
- "Dynamic Meshing Using Adaptively Sampled Distance Fields", SIGGRAPH 2001 Conference Abstracts and Applications
- "New Directions in Shape Representations", SIGGRAPH 2001 (full day) Course
- "Using Distance Maps for Accurate Surface Representation in Sampled Volumes", IEEE VolVis Symposium 1998
- "A New Representation for Device Color Gamuts", MERL TR2001-09
- "A New Framework For Non-Photorealistic Rendering", MERL TR2001-12
- "A New Interaction Method for Creating and Editing 3D Geometry and Geometric Texture", SIGGRAPH 2002 Submission

Patents

- A comprehensive patent portfolio
  - 2 issued patents
  - 3 granted patents (but not yet issued)
  - 17 filed patent applications
  - 4 new patent disclosures

ADF Library

- A product-worthy C library
  - Features include: Stock distance functions for constructing and combining objects; Milling specific distance functions for extrusion, surface of revolution, and lathing; Tiled generation; Bounded-surface generation; Interactive CSG editing; Bezier tool paths; Surface and volume rendering; Procedural shading interface; Adaptive, asynchronous ray casting; ADF specific 2D antialiasing; Supersampling for standard 2D and 3D antialiasing; Simple camera and lighting model; Region rendering to support interactive CSG editing; Conversion of image and range data to ADFs; Idle time processing; Reconstruction functions; ADF read and write operations; Interactive generation of view-dependent and view-independent point models; Interactive generation of optimal triangle meshes; Generation of level-of-detail triangle meshes; Blending of ADFs; Input and output of Wavefront Object files; Amenable to parallel implementations; Developed with object-oriented ANSI C; Runs under Windows and Linux.

Demonstration

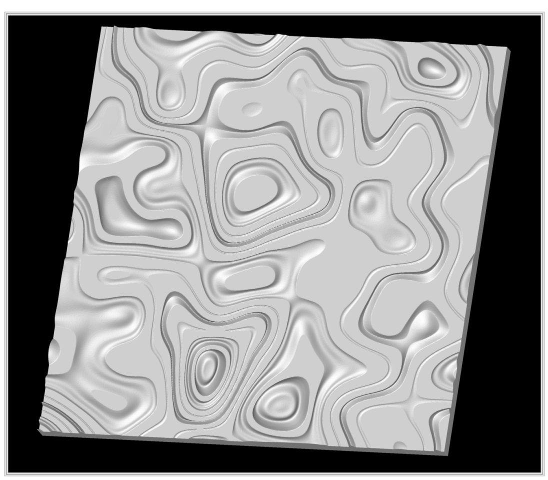
Business Opportunities

- Digital clay
- Conceptual design
- Real-time simulation, verification, and path planning for NC milling

Digital Clay

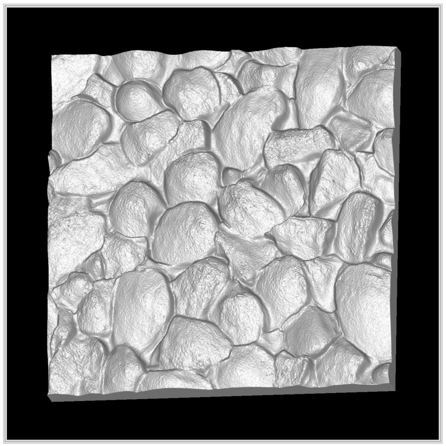
- ADFs provide a fresh approach to design with
  - direct sculpting interface
  - organic shapes
  - razor sharp edges
  - highly detailed texture from range images, photographs, and procedurally generated data
- Market opportunities include
  - constructing Hollywood models
  - constructing models for game design
  - 3D ‘sketching’ for industrial design

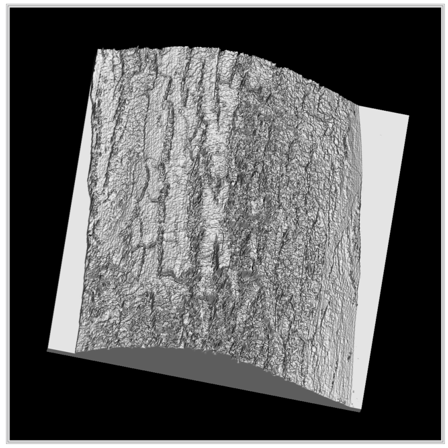
Organic and Textured Sculpting



Organic shape with razor sharp edges

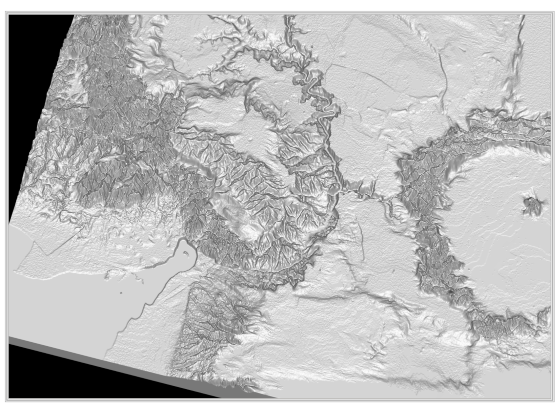
Hollywood Models





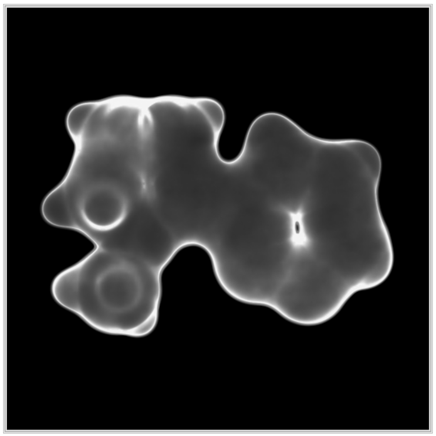
Exquisitely detailed concept models for “The Lord of the Rings”
(simple Phong illumination – all detail is geometric)

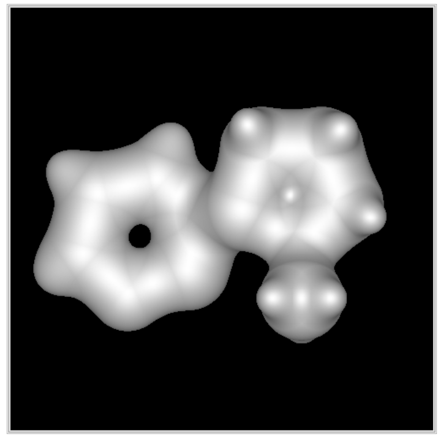
Hollywood Models



Concept model of Middle-earth for “The Lord of the Rings”
(simple Phong illumination – all detail is geometric)

Organic and Textured Sculpting



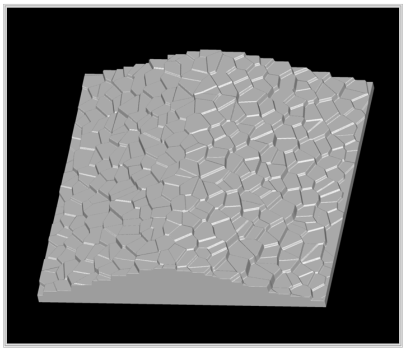


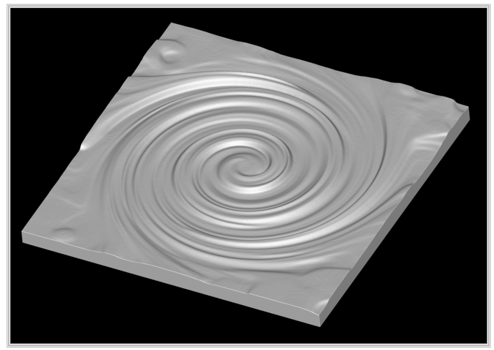
Organic forms

Digital Clay

- Advantages of ADFs for Editing
  - Represent both smooth surfaces and sharp corners without excessive memory
  - Sculpting is direct, intuitive, and fast
  - Does not require control point manipulation or trimming
  - The distance field can be used to enhance the user interface
    - Guide the position and orientation of the sculpting tool
    - Enable distance-based constraints for carving
  - ADF-specific methods for capturing geometry from range data and photographs

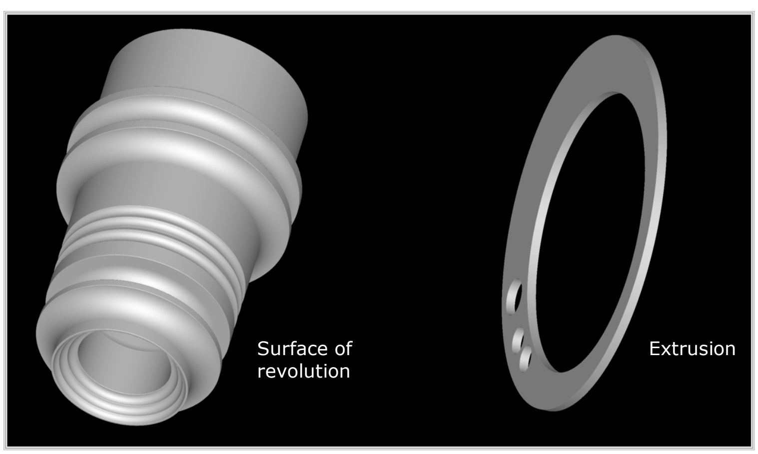
Conceptual Design





ADF Concept models

Conceptual Design



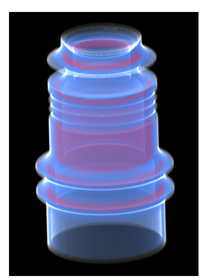
3D ADFs generated directly from sculpted 2D ADFs

Conceptual Design

- Reverse engineering from range data
  - fast and memory efficient
  - water-tight, hole-free models
  - can be trivially sculpted in 3D to repair occluded regions
  - can produce optimal level-of-detail tessellations

NC Milling

- Real-time simulation
  - Fast editing rates
  - Accurate shape representation
- Verification, Analysis, and Path Planning
  - Distance field enables fast and accurate error measurement
  - Trivial collision detection and proximity testing between tool and workpiece
  - ADFs represent surfaces, object interiors, and the material to be removed
  - Offset surfaces can be used for rough cutting in coarse-to-fine machining
  - Volume visualization for part thickness testing



Red: thickness > 0.02

The End

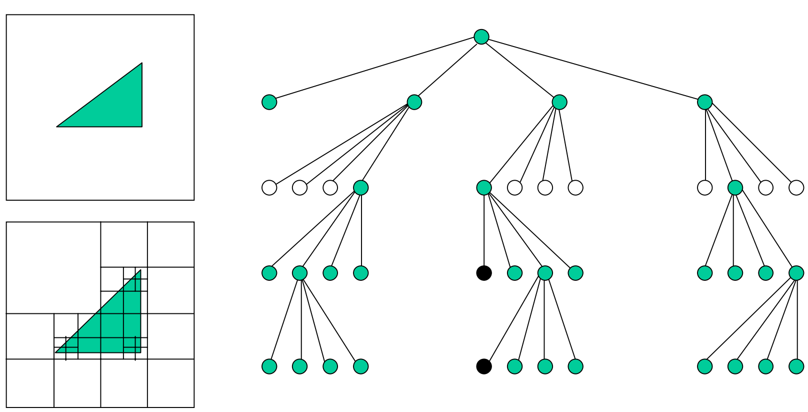
Euclidean and Non-Euclidean Fields

- Consider the distance field of the unit sphere S in $\mathbf{R}^3$ given by $h(\mathbf{x}) = 1 - (x^2 + y^2 + z^2)^{1/2}$, in which h is the **Euclidean signed distance** from S
- Or $h(\mathbf{x}) = 1 - (x^2 + y^2 + z^2)$, in which h is the **algebraic signed distance** from S
- Or $h(\mathbf{x}) = (1 - (x^2 + y^2 + z^2))^2$, in which h is an **unsigned distance** from S
- Etc.

Shape

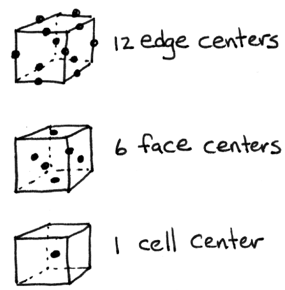
- We use it in a broad context for **any locus defined in a metric space**
  - Locus: any system of points which satisfies one or more conditions
  - Metric space: a pair (X, d) where X is a set and d is a metric on X such that
    - d(x,y) ≥ 0 for all x,y in X
    - d(x,y) = 0 iff x = y for all x,y in X
    - d(x,y) = d(y,x) for all x,y in X
    - d(x,z) ≤ d(x,y) + d(y,z) for all x,y,z in X

Example 2D Quadtree ADF



Subdivision Predicates

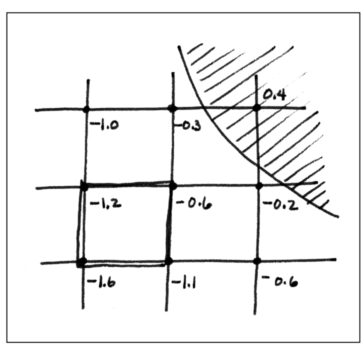
- Point sampling
- Gradient sampling
- Interval methods



Point sampling: 19 test points
to determine cell error

Euclidean ADFs

- Can efficiently determine if a cell is interior or exterior



(1) all d_i have same sign
(2) all $|| d_i || > 0.5$ cell diagonal