Course Roadmap

Rasterization Pipeline
Core Concepts
- Sampling
- Antialiasing
- Transforms

Intro
Rasterization
Transforms & Projection
Texture Mapping
Visibility, Shading, Overall Pipeline

Starting today

Geometric Modeling

Lighting & Materials

Cameras & Imaging
Lecture 8:
Introduction to Geometry

Computer Graphics and Imaging
UC Berkeley CS184/284A
Examples of Geometry
Examples of Geometry

NetCarShow.com

CS184/284A Ren Ng
Examples of Geometry
Examples of Geometry
Examples of Geometry
Examples of Geometry
Examples of Geometry
Examples of Geometry
Examples of Geometry
Examples of Geometry
No “Best” Representation – Geometry is Hard!

“I hate meshes.
I cannot believe how hard this is.
Geometry is hard.”

— David Baraff
Senior Research Scientist
Pixar Animation Studios
Many Ways to Represent Geometry

Explicit
- point cloud
- polygon mesh
- subdivision, NURBS
- ...

Implicit
- level sets
- algebraic surface
- distance functions
- ...

Each choice best suited to a different task/type of geometry
“Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

E.g. sphere: all points in 3D, where \( x^2 + y^2 + z^2 = 1 \)

More generally, \( f(x,y,z) = 0 \)
Implicit Surface – Sampling Can Be Hard

\[ f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1 \]

What points lie on \( f(x, y, z) = 0 \)?

Some tasks are hard with implicit representations
Implicit Surface – Inside/Outside Tests Easy

\[ f(x, y, z) = x^2 + y^2 + z^2 - 1 \]

Is \((3/4, 1/2, 1/4)\) inside?

Just plug it in:

\[ f(x,y,z) = -1/8 < 0 \]

Yes, inside.

Implicit representations make some tasks easy.
“Explicit” Representations of Geometry

All points are given directly

Generally:

\[ f : \mathbb{R}^2 \rightarrow \mathbb{R}^3; (u, v) \mapsto (x, y, z) \]

(Might have multiple maps, e.g., one per triangle.)
Explicit Surface – Sampling Is Easy

\[ f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u) \]

What points lie on this surface?

Just plug in \((u, v)\) values!

Explicit representations make some tasks easy.
Explicit Surface – Inside/Outside Test Hard

\[ f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v) \]

Is \((3/4, 1/2, 1/4)\) inside?

Some tasks are hard with explicit representations
Best Representation Depends on the Task!
Attendance Time

If you are seated in class, go to this form and sign in:

- https://tinyurl.com/184lecture

Notes:

- Time-stamp will be taken when you submit form. Do it now, won’t count later.
- Don’t tell friends outside class to fill it out now, because we will audit at some point in semester.
- Failing audit will have large negative consequence. You don’t need to, because you have an alternative!
Implicit Representations in Computer Graphics
Many Implicit Representations in Graphics

Algebraic surfaces
Constructive solid geometry
Level set methods
Blobby surfaces
Fractals

...
Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in $x$, $y$, $z$

\[
x^2 + y^2 + z^2 = 1
\]

\[
(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2
\]

\[
(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 = x^2z^3 + \frac{9y^2z^3}{80}
\]

More complex shapes?
Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations

Boolean expressions:
Blobby Surfaces (Implicit)

Instead of Booleans, gradually blend surfaces together:

Easier to understand in 2D:

\[
\phi_p(x) := e^{-|x-p|^2} \quad \text{(Gaussian centered at } p) \\
f := \phi_p + \phi_q \quad \text{(Sum of Gaussians at different centers)}
\]

\[f = \frac{1}{2}\]
Blending Distance Functions (Implicit)

A distance function gives minimum distance to object

Can blend any two distance functions $d_1$, $d_2$:

Similar strategy to points, though many possibilities. E.g.,

$$f(x) := e^{d_1(x)^2} + e^{d_2(x)^2} - \frac{1}{2}$$
Scene of Pure Distance Functions (Not Easy!)

Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting)
But, hard to describe complex shapes in closed form
Alternative: store a grid of values approximating function

Surface is found where interpolated values equal zero
Provides much more explicit control over shape (like a texture)
Level Sets from Medical Data (CT, MRI, etc.)

Level sets encode, e.g., constant tissue density
Level Sets in Physical Simulation

Level set encodes distance to air-liquid boundary

See http://physbam.stanford.edu
Fractals (Implicit)

Exhibit self-similarity, detail at all scales
“Language” for describing natural phenomena
Hard to control shape!
Mandelbrot Set - Definition

For each point $c$ in the plane:

- double the angle
- square the magnitude
- add the original point $c$
- repeat

If the point remains bounded (never goes to $\infty$), it's in the set.
Mandelbrot Set - Examples

Starting point

- $(0, 1/2)$ (converges)
- $(0, 1)$ (periodic)
- $(1/3, 1/2)$ (diverges)
Mandelbrot Set - Zooming In

(Colored according to how quickly each point diverges/converges.)
Implicit Representations - Pros & Cons

Pros:

• description can be very compact (e.g., a polynomial)
• certain queries easy (inside object, distance to surface)
• good for ray-to-surface intersection (more later)
• for simple shapes, exact description / no sampling error
• easy to handle changes in topology (e.g., fluid)

Cons:

• hard to find all points in shape (e.g., for drawing)
• difficult to model complex shapes
Explicit Representations in Computer Graphics
Many Explicit Representations in Graphics

triangle meshes
bezier surfaces
subdivision surfaces
NURBS
point clouds
...

(Will spend most of our time on a few of these.)
Point Cloud (Explicit)

Easiest representation: list of points \((x, y, z)\)

Often augmented with normals

Easily represent any kind of geometry

Useful for LARGE datasets (\(>>1\) point/pixel)

Difficult to draw in undersampled regions
Polygon Mesh (Explicit)

Store vertices & polygons (often triangles or quads)
Easier to do processing/simulation, adaptive sampling
More complicated data structures
Perhaps most common representation in graphics
Bézier Curves (Explicit)

Bézier curves

γ(s)

Piecewise Bézier
Bézier Surfaces (Explicit)

Use tensor product of Bézier curves to get a patch:

Multiple Bézier patches form a surface
Subdivision Curves (Explicit)

Alternative starting point for smooth curves: subdivision
Start with control polygon
Insert new vertex at each edge midpoint
Update vertex positions according to fixed rule
For careful choice of averaging rule, yields smooth curve
  • Corner-cutting (Chaikin): quadratic B-spline
Subdivision Surfaces (Explicit)

Start with coarse polygon mesh ("control cage")
Subdivide each element
Update vertices via local averaging
Many possible rules:
- Catmull-Clark (quads)
- Loop (triangles)
- ...
Common issues:
- interpolating or approximating?
- continuity at vertices?
Relatively easy for modeling; harder to guarantee continuity
Things to Remember

Implicit and explicit representations of geometry

- Different tasks are easy/hard with implicit/explicit
- Many implicit and explicit representations

Implicit examples

- Algebraic formula encoding a distance function

Explicit examples

- Triangle and quad meshes
- Bezier curves and surfaces
- Subdivision curves and surfaces
Coming Up

Next lecture: smooth curves and surfaces
  • Splines and Bézier curves
  • Bicubic surfaces

Lectures after that:
  • Mesh representations
  • Mesh upsampling, downsampling, resampling
  • Geometric queries, e.g. ray-surface intersection
  • Accelerating geometric queries
Acknowledgments

Thanks to Keenan Crane, who provided the majority of these slides. Thanks also to Steve Marschner, Pat Hanrahan, and Mark Pauly.