Geometric Modeling
Summer Semester 2010

Introduction
Motivation · Topics · Basic Modeling Techniques
Today...

Topics:

• Formalities & Organization
• Introduction: Geometric Modeling
• Mathematical Tools (1)
Topics:

- Formalities & Organization
- Introduction: Geometric Modeling
  - Motivation
  - Overview: Topics
  - Basic modeling techniques
- Mathematical Tools (1)
Motivation
Motivation

This lecture covers two related areas:

- Classic geometric modeling
- Geometry processing

Common techniques (math, models, terminology), but different goals
Geometric Modeling

Geometric Modeling:

• You start with a blank screen, design a geometric model

• Typical techniques:
  ▪ Triangle meshes
  ▪ Constructive Solid Geometry (CSG)
  ▪ Spline curves & surfaces
  ▪ Subdivision surfaces

• Goal is *interactive modeling*

• Mathematical tools are designed *with the user in mind*
Geometry Processing

You already have a geometric model
- Typically from a 3D range scanner (read: not nice)
- You need to process & edit the geometry
- The original model has not been build with the user in mind (stupid range scanner)

Typical techniques:
- Noise removal, filtering
- Surface reconstruction
- Registration
- Freeform deformation modeling
- Statistical analysis (features, symmetry, hole-filling etc...)
Our Perspective

The perspective of this lecture:

- The basic mathematical tools for handling geometry are the same
- Different usage, adaptation, specific algorithms
- We will discuss
  - The basic concepts and tools (mathematical foundation, representations, basic algorithms)
  - ...and applications in both areas.
Examples:

Geometric Modeling
The Modern World...

designed on a computer
(the building)

designed on a computer as well
(the cars)

fortunately, not (yet) designed
on a computer
(the trees)

(c.f. Danny Hillis, Siggraph 2001 keynote)
Impact of Geometric Modeling

We live in a world designed using CAD

- Almost any man-made structure is nowadays planed and designed using computers
  - Architecture
  - Commodities: Chairs, furniture, your microwave & toaster
  - Your car (in case you have one, but probably the bike as well)
    - spline curves have actually been invented in the automotive industry
  - Typesetting

- <advertising> Our abilities in geometric modeling shapes the world we live in each day. </advertising>
Different Modeling Tasks

CAD / CAM

- Precision Guarantees
- Handle geometric constraints exactly (e.g. exact circles)
- Modeling guided by rules and constraints
Different Modeling Tasks

Photorealistic Rendering

- Has to “look” good
- Ad-hoc techniques are ok
- Using textures & shaders to “fake” details
- More complexity, but less rigorous

Just two examples, lots of stuff in between...
Examples:

Geometry Processing
Geometry Processing

A rather new area

- Motivation: 3D scanning
  - You (your company) can buy devices that scan real world 3D objects
  - You get (typically) clouds of measurement points
- Many other sources of geometry as well:
  - Science (CT, [F]MRI, ET, Cryo-EM, ...)
  - 3D movie making
  - The design department of your company has dozens of TByte of “polygon soup”...
  - Crawl the internet
- Need to process the geometry further
Photoshopping Geometry

Geometry Processing:

- **Cleanup:**
  - Remove inconsistencies
  - Make watertight (well defined inside/outside, for 3D printers)
  - Simplify – keep only the main “structure”
  - Remove noise, small holes, etc...

- **Touch-up /Edit:**
  - Texturing, painting, carving
  - Deformation
  - Stitch together pieces

- **Lots of other stuff – similar to image processing**
Example

Example: The Stanford “Digital Michelangelo Project”

[Levoy et al.: The Digital Michelangelo Project, Siggraph 2000]
Scan Registration

[data set: Stanford 3D Scanning Repository]
Feature Tracking

Fully Automatic:

[Implementation: Martin Bokeloh (Diploma thesis)]
Scanning the World....

Example: The “Wägele”

[Laser scanners (2D sheets of distance measurements)]

[Biber et al. 2005]

A pull-through measurement device – can acquire complete buildings in a few hours
This is what you get...

Corridor – CS Building
University of Tübingen (6.5 GB)

...lots of artifacts
(the scanner does not really like windows)

CS Building, Outside
(nicer colors...)
Automatic Processing

Example: Automatic Outlier Removal
Think Big

More Problems:

- Occluded areas, shiny / transparent objects
  \(\Rightarrow\) holes (lots of holes, actually)
- Huge amounts of data (really huge)

City Scanning

- There are big companies trying to scan large areas
- Think Google Earth in full resolution
- How about a virtual online walk through
  \textit{New York, Tokyo, Saarbrücken}?
- Lots of open research problems to get there
HUGE Data Sets

The Largest Data Set I have On My Hard-Drive...

Data set: Outdoor Scan (structure from video) of a part of the UNC campus ($2.2 \cdot 10^9$ pts / 63.5 GB), courtesy of J.-M. Frahm, University of North Carolina
Hole Filling

Wei-Levoy Texture Synthesis Algorithm:

[Implementation: Alexander Berner (Diploma thesis)]
Filling Holes

[implementation: Alexander Berner (Diploma thesis)]
Filling Holes

[implementation: Alexander Berner (Diploma thesis)]
Symmetry Detection

[data set: M. Wacker, HTW Dresden]
Results
Results
Results
Line Feature Matching

[data sets: Kartographisches Institut, Universität Hannover / M. Wacker, HTW Dresden]
Reconstruction by Symmetry

overlay of 16 parts

[data sets: Kartographisches Institut, Universität Hannover]
Overview

Our approach

- Take *existing* model
- Analyse shape structure
- Derive shape modification rules
Technique Overview

Conceptual Steps:

• Symmetry detection
• Finding *docking sites* and *dockers*
• Combine into a *shape grammar*
Results
Results

~500,000 triangles
Results
Results
Deformable Shape Matching

[data set: Stanford 3D Scanning Repository]
Problem Statement

Deformable Matching

- Two shapes: original, deformed
- How to establish correspondences?
- Looking for global optimum
  - Arbitrary pose

Assumption

- Approximately isometric deformation

[data set: S. König, TU Dresden]
Results

[data sets: Stanford 3D Scanning Repository / Carsten Stoll]
Animation Reconstruction

[data set: P. Phong, Stanford University]
Real-time 3D scanners:

- Acquire geometry at video rates
- Capture 3D movies: “performance capture”
- Technique still immature, but very interesting applications, in particular special effects for movies

[Davis et al. 2003]
Real-Time 3D Scanners

- **space-time stereo**
courtesy of James Davis
University of California at Santa Cruz

- **color-coded structured light**
courtesy of Phil Fong
Stanford University

- **high-speed structured light**
courtesy of Stefan Gumhold
TU Dresden
Reconstruction

Dynamic geometry reconstruction

- Hole filling
- Remove noise and outliers
- Establish correspondences
  - Need to know where every point on the object goes to over time
  - Simplifies further editing
Animation Reconstruction

Remove noise, outliers

Fill-in holes
(from all frames)

Dense correspondences
Factorization

\[ f(x, t) \] – deformation field

\[ t = 0 \quad t = 1 \quad t = 2 \]

\[ x \] – point on urshape \( S \)

\[ d_{t,i} \] – data points

[data set courtesy of P. Phong, Stanford University]
79 frames, 24M data pts, 21K surfels, 315 nodes
Overview

Topics
Overview: Geometric Modeling 2010

Mathematical Background (Recap)

• Linear algebra: vector spaces, function spaces, quadrics
• Analysis: multi-dim. calculus, differential geometry
• Numerics: quadratic and non-linear optimization

Geometric Modeling

• Smooth curves: polynomial interpolation & approximation, Bezier curves, B-Splines, NURBS
• Smooth surfaces: spline surfaces, implicit functions, variational modeling
• Meshes: meshes, multi-resolution, subdivision
Overview: Geometric Modeling 2010

Geometry Processing

- 3D Scanning: Overview
- Registration: ICP, NDT
- Surface Reconstruction: smoothing, topology reconstruction, moving least-squares
- Editing: free-form deformation

Preliminary List:

- Topics might change
- Not presented strictly in this order
Current List of Topics (subject to changes):

- Math Background:
  - Linear Algebra, Analysis, Differential Geometry, Numerics, Topology
- Interpolation and Approximation
- Spline Curves
- Blossoming and Polar Forms
- Rational Splines
- Spline Surfaces
- Subdivision Surfaces
- Implicit Functions
- Variational Modeling
- Point Based Representations
- Multi Resolution Representations
- Surface Parametrization
Overview
Modeling Techniques
Geometric Modeling

What do we want to do?

Geometric object

$B \subseteq \mathbb{R}^3$

Empty space

(typically $\mathbb{R}^3$)
Fundamental Problem

The Problem:

\[ \mathbb{R}^d \]

\[ B \]

*infinite* number of points

*my computer*: 8GB of memory

We need to encode a continuous model with a finite amount of information
Modeling Approaches

Two Basic Approaches

- Discrete representations
  - Fixed discrete bins
- “Continuous” representations
  - Mathematical description
  - Evaluate continuously
You know this...

- **Fixed Grid of values:**
  \[(i_1, \ldots, i_{d_s}) \in \mathbb{Z}^{d_s} \rightarrow (x_1, \ldots, x_{d_t}) \in \mathbb{R}^{d_t}\]

- **Typical scenarios:**
  - \(d_s = 2, d_t = 3\): Bitmap images
  - \(d_s = 3, d_t = 1\): Volume data (scalar fields)
  - \(d_s = 2, d_t = 1\): Depth maps (range images)

- **PDEs:** “Finite Differences” models
Modeling Approaches

Two Basic Approaches

- Discrete representations
  - Fixed discrete bins
- “Continuous” representations
  - Mathematical description
  - Evaluate continuously
Continuous Models

**Basic Principle: Procedural Modeling**

- **Query Parameters**
  (a finite set of numbers from a continuous set)

- **Algorithm(s)**
  determines the *class of objects* that can be represented

- **Answer**

- **finite set of Shape Parameters**
  determines the object *shape*
Example: Continuous Model

Example: Sphere

• Shape Parameters: center, radius (4 numbers)

• Algorithms:

  ▪ **Ray Intersection** (e.g. for display)
    – Input: Ray (angle, position: 5 numbers)
    – Output: {true, false}

  ▪ **Inside/outside test** (e.g. for rasterization)
    – Input: Position (3 numbers)
    – Output: {true, false}

  ▪ **Parametrization** (e.g. for display)
    – Input: longitude, latitude ($\alpha$, $\beta$)
    – Output: position (3 numbers)
So Many Questions...

Several algorithms for the same representation:

- **Parametrization** – compute surface points according to continuous parameters
- *(Signed)* distance computation – distance to surface of points in space, inside/outside test
- **Intersection** – with rays (rendering), other objects (collision detection)
- **Conversion** – into other representations.
- Many more...

In addition, we also need algorithms to construct and alter the models.
Continuous, Procedural Models

“Continuous” representations

- An algorithm describes the shape
- The shape is determined by a finite number of continuous parameters
- The shape can be queried with a finite number of continuous parameters
- More involved (have to ask for information)
- But potentially “infinite” resolution (continuous model)
- Structural model complexity still limited by algorithm and parameters

This lecture examines these representations and the corresponding algorithms
Classes of Models

(Main) classes of models in this lecture:

- Primitive meshes
- Parametric models
- Implicit models
- Particle / point-based models

Remarks

- Most models are hybrid (combine several of these)
- Representations can be converted (may be approximate)
- Some questions are much easier to answer for certain representations
Modeling Zoo

- Parametric Models
- Implicit Models
- Primitive Meshes
- Particle Models
Modeling Zoo

Parametric Models

Implicit Models

Primitive Meshes

Particle Models
Parametric Models

- Function $f$ maps from parameter domain $\Omega \subseteq \mathbb{R}^d$ to target space $S \subseteq \mathbb{R}^t$
- Evaluation of $f$ gives one point on the model $(u, v)$
<table>
<thead>
<tr>
<th>Input</th>
<th>Output: 1D</th>
<th>Output: 2D</th>
<th>Output: 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td><img src="image" alt="Function Graph" /></td>
<td><img src="image" alt="Plane Curve" /></td>
<td><img src="image" alt="Space Curve" /></td>
</tr>
<tr>
<td>2D</td>
<td><img src="image" alt="Plane Warp" /></td>
<td><img src="image" alt="Surface" /></td>
<td><img src="image" alt="Space Warp" /></td>
</tr>
<tr>
<td>3D</td>
<td><a href="image">Image</a></td>
<td><a href="image">Image</a></td>
<td><a href="image">Image</a></td>
</tr>
<tr>
<td>Input: 1D</td>
<td>Output: 1D</td>
<td>Input: 2D</td>
<td>Output: 2D</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td><img src="image1" alt="Function Graph" /></td>
<td><img src="image2" alt="Plane Curve" /></td>
<td><img src="image3" alt="Plane Warp" /></td>
<td><img src="image4" alt="Surface" /></td>
</tr>
</tbody>
</table>
Modeling Zoo

Parametric Models

Implicit Models

Primitive Meshes

Particle Models
Primitive Meshes

- Collection of geometric primitives
  - Triangles
  - Quadrilaterals
  - More general primitives (spline patches)

- Typically, the primitives are parametric surfaces

- Composite model:
  - Mesh encodes topology, rough shape
  - Primitive parameter encode local geometry

- Triangle meshes rule the world (“triangle soup”)
Complex Topology for Parametric Models

- Mesh of parameter domains attached in a mesh
- Domain can have complex shape ("trimmed patches")
- Separate mapping function $f$ for each part (typically of the same class)
Meshes are Great

Advantages of mesh-based modeling:

• Compact representation (usually)
• Can represent arbitrary topology
• Using the right parametric surfaces as parts, many important geometric objects can be represented exactly (e.g. NURBS: circles, cylinders, spheres $\rightarrow$ CAD/CAM)
Meshes are not so great

Problem with Meshes:

• Need to specify a mesh first, then edit geometry

• Problems for larger changes
  ▪ Mesh structure and shape need to be adjusted
  ▪ Mesh encodes object topology
    ➞ Changing object topology is painful

• Difficult to use for many applications (such as surface reconstruction)
  ▪ Rule of thumb: If the topology or the coarse scale shape changes drastically and frequently during computations, meshes are hard to use
  ▪ Drastic example: Fluid simulation (surface of splashing water)
Modeling Zoo

Parametric Models

Implicit Models

Primitive Meshes

Particle Models
Implicit Modeling

General Formulation:

• Curve / Surface $S = \{x \mid f(x) = 0\}$
• $x \in \mathbb{R}^d$ ($d = 2,3$), $f(x) \in \mathbb{R}$
• $S$ is (usually) a $d$-1 dimensional object

This means...:

• The surface is obtained implicitly as the set of points for which some given function vanishes ($f(x) = 0$)
• Alternative notation: $S = f^{-1}(0)$ (“inverse” yields a set)
Implicit Modeling

Example:

• Circle: \( x^2 + y^2 = r^2 \)
  \[ \iff f_r(x,y) = x^2 + y^2 - r^2 = 0 \]

• Sphere: \( x^2 + y^2 + z^2 = r^2 \)

Special Case:

• Signed distance field

• Function value is signed distance to surface

\[ f(x,y) = \text{sign}(x^2 + y^2 - r^2)\sqrt{|x^2 + y^2 - r^2|} \]

• Negative means inside, positive means outside
Implicit Modeling

Example:
- Circle: $x^2 + y^2 = r^2$
  $\iff f_r(x,y) = x^2 + y^2 - r^2 = 0$
- Sphere: $x^2 + y^2 + z^2 = r^2$

Special Case:
- Signed distance field
- Function value is signed distance to surface
  $f(x,y) = sign(x^2 + y^2 - r^2) \sqrt{|x^2 + y^2 - r^2|}$
- Negative means inside, positive means outside

“Signed squared distance field”
(has some useful properties, e.g. from a statistical point of view)
Implicit Modeling: Pros & Cons

Advantages:

- More general than parametric techniques
- Topology can be changed easily (depends on how \( f \) is specified, though)
- Implicit representations are the standard technique for simulations with \textit{free boundaries}. Also known as “\textit{level-set methods}”.
- Typical example: Fluid simulation (evolving water-air interface)
- Geometric modeling: Surface reconstruction, “blobby surfaces”
Implicit Modeling: Pros & Cons

Disadvantages:

- Need to solve inversion problem $S = f^{-1}(0)$
- Algorithms for display, conversion etc. tend to be more difficult and more expensive (inside/outside test is easy though for signed distance fields)
- Representing objects takes more memory (we will discuss standard representations later)
Modeling Zoo

Parametric Models

Implicit Models

Primitive Meshes

Particle Models
Particle / Point-based Representations

- Geometry is represented as a set of points / particles
- The particles form a (typically irregular) *sample* of the geometric object
- Need additional information to deal with “the empty space around the particles”

*additional assumptions*
Particle Representations

Helpful Information

• Each particle may carry a set of attributes
  ▪ Must have: Its position
  ▪ Additional geometric information: Particle density (sample spacing), surface normals
  ▪ Additionally: Color, physical quantities (mass, pressure, temperature), ...

• This information can be used to improve the reconstruction of the geometric object described by the particles
The Wrath of Khan

Why Star Trek is at fault...

- Particle methods first used in computer graphics to represent fuzzy phenomena (fire, clouds, smoke)
- “Particle Systems—a Technique for Modeling a Class of Fuzzy Objects” [Reeves 1984]
- Probably most well-known example: Genesis sequence
Genesis Sequence [Reeves 1983]
Non-Fire Objects

Particle Traces for Modeling Plants
(also from [Reeves 1983])
Geometric Modeling

How became the geometric modeling crowd interested in this?

3D Scanners

- 3D scanning devices yield point clouds (often: measure distance to points in space, one at a time)
- Then you have to deal with the problem anyway
- Need algorithms to directly work on “point clouds” (this is the geometry name for particle system)
Geometric Modeling

How became the geometric modeling crowd interested in this?

Other Reasons:

• Similar advantages as implicit techniques
• Topology does not matter (for the good and for the bad)
  ▪ Topology is easy to change
  ▪ Multi-scale representations are easy to do
    (more details on multi-resolution techniques later)
• Often easier to use than implicit or parametric techniques
Multi-Scale Geometry w/Points
Summary

- Lots of different representations
- No silver bullet
- In theory, everything always works, but might be just too complicated/expensive
- Best choice depends on the application
- We will look on all of this...
  - Focus on parametric techniques though
  - Most common approach