Geometric Modeling

Summer Semester 2012

Introduction

Motivation · Topics · Basic Modeling Techniques

UNIVERSITÄT DES SAARLANDES

CLUSTER OF EXCELLENCE

max planck institut informatik

Today...

Topics:

- Formalities & Organization
- Introduction: Geometric Modeling
  - Motivation
  - Overview: Topics
  - Basic modeling techniques
- Mathematical Background
  - Function Spaces
  - Differential Geometry

Today...

Topics:

- Formalities & Organization
- Introduction: Geometric Modeling
  - Motivation
  - Overview: Topics
  - Basic modeling techniques
- Mathematical Background
  - Function Spaces
  - Differential Geometry

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

Geometry Processing

- You already have a geometric model
  - Typically: 3D scanner (read: not nice)
  - You need to process & edit the geometry
  - Complex, unstructured data
- Typical problems
  - Noise removal, filtering
  - Surface reconstruction
  - Registration
  - Statistical analysis (features, symmetry, hole-filling etc...)
  - Structure retrieval / data interpretation

Two fields

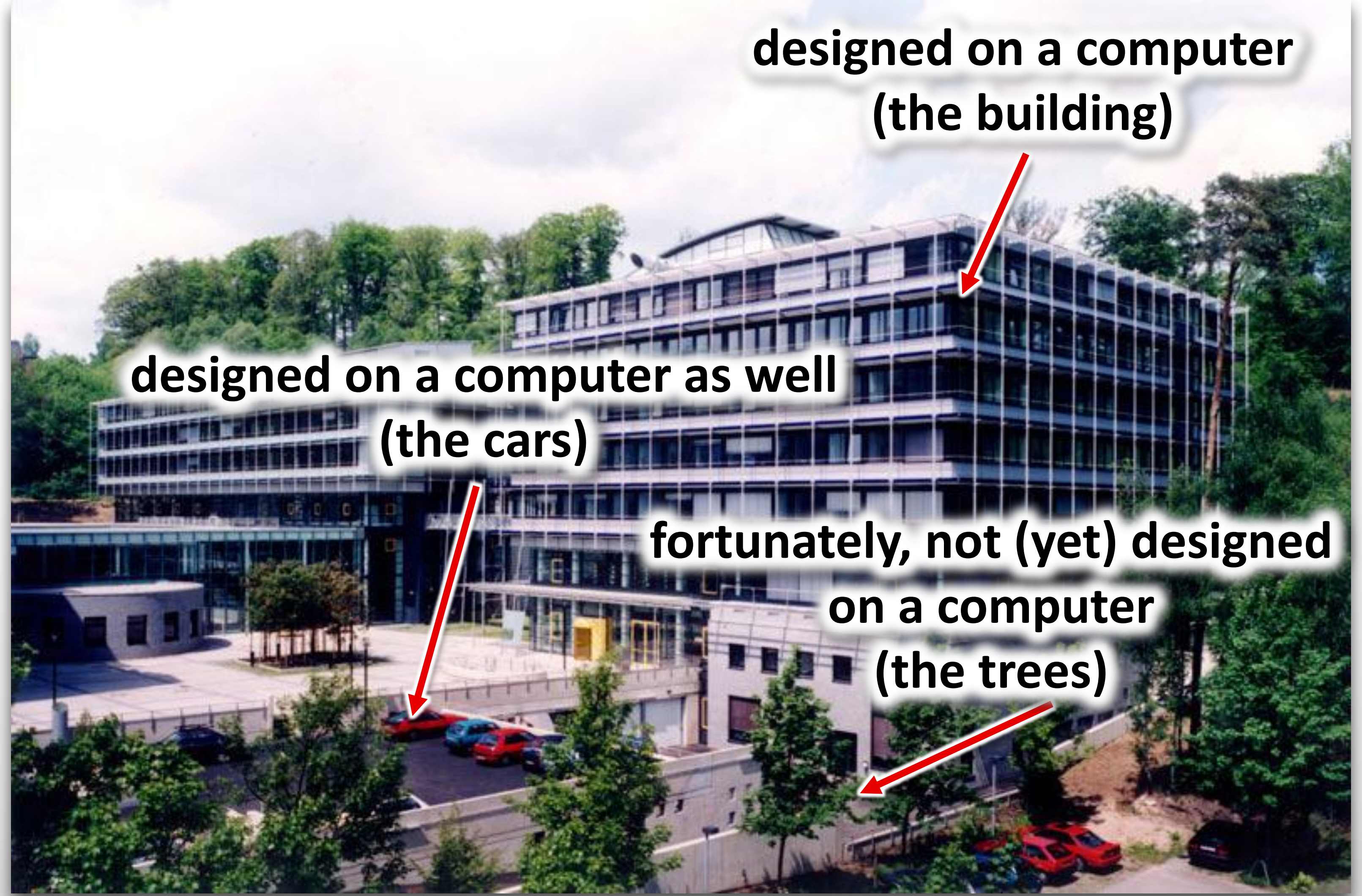
Goals

- Geometric modeling
  - Create nice images
  - Build stuff
  - Long term goal: *model anything*
- Geometry processing
  - Data processing (think "Photoshop")
  - Reverse engineering
  - Long term goal: *shape understanding*
    machines make sense
    of the world

Examples:

Geometric Modeling

The Modern World...



[c.f. Danny Hillis, Siggraph 2001 keynote]

Impact of Geometric Modeling

We live in a world designed using CAD

- Almost any man-made structure designed w/computers
  - Architecture
  - Commodities
  - Your car (bikes as well)
  - Spline curves invented in 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

Examples:

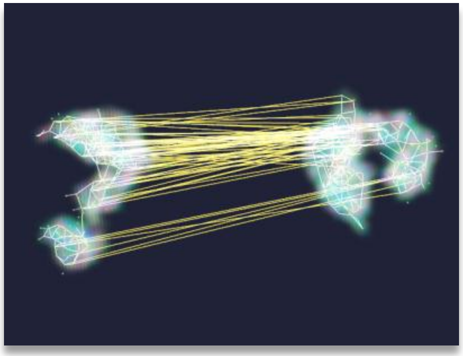
Geometry Processing

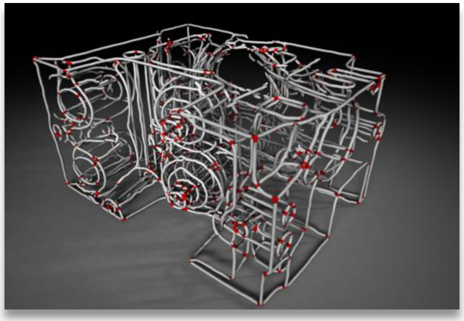
Geometry Processing

A rather new area

- Motivation: 3D scanning
  - 3D scanners
  - Clouds of millions of measurement points

- Sources of spatial data:
  - Science: CT, [F]MRI, ET, Cryo-EM, ...
  - 3D movie making
  - Game / movie industry:
    Servers with GBs of “polygon soup”
  - Crawl the internet





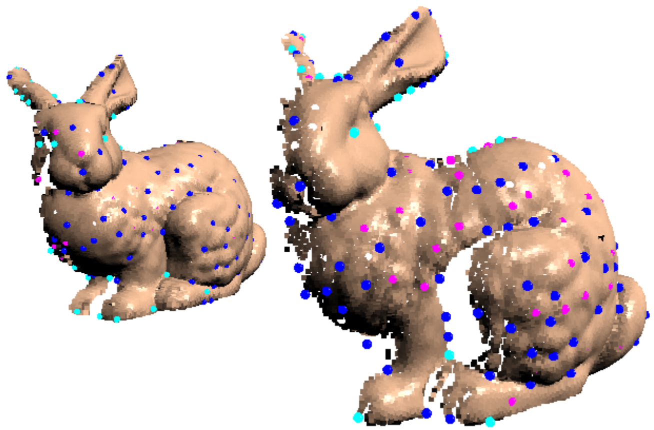
- Need to process the geometry further

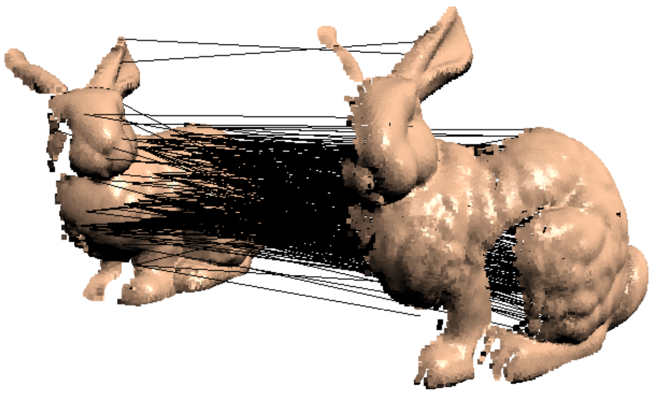
Photoshopping Geometry

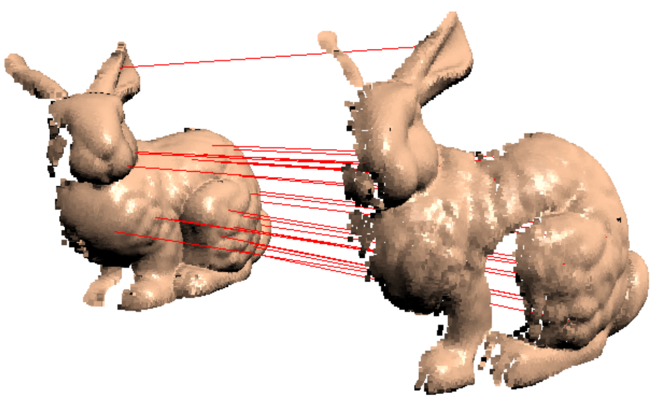
Geometry Processing:

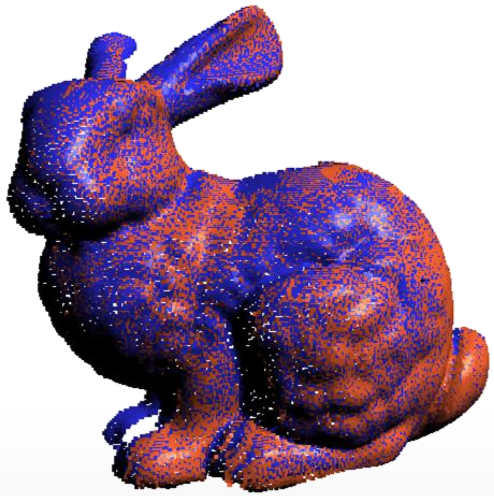
- Cleanup:
  - Remove inconsistencies
  - Make watertight (well defined inside/outside, for 3D printers)
  - Simplify – keep only the main "structure"
  - Remove noise, small holes, etc...
- Touch-up /Edit:
  - Texturing, painting, carving
  - Deformation
  - Stitch together pieces
- Lots of other stuff – similar to image processing

Scan Registration





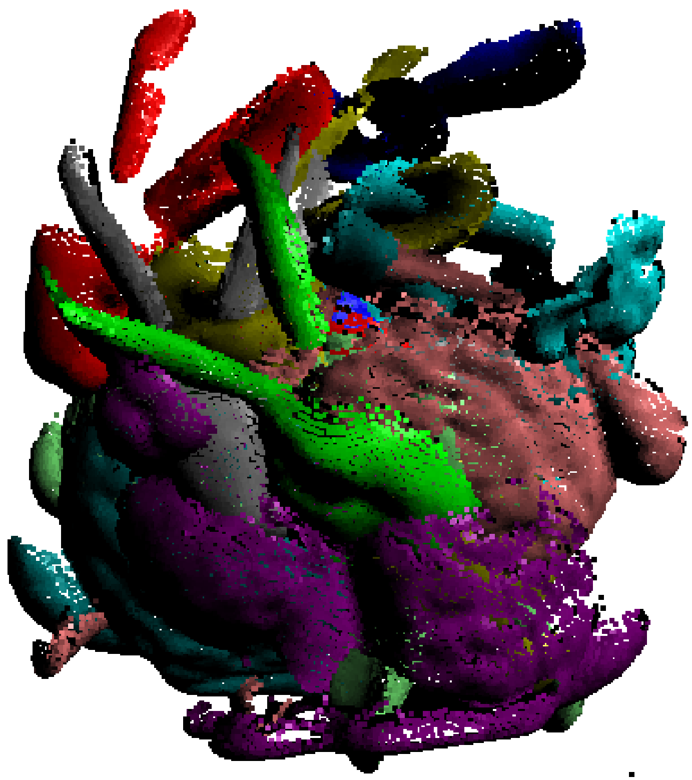




[data set: Stanford 3D Scanning Repository]

Feature Tracking

Fully Automatic:



[Implementation: Martin Bokeloh (Diploma thesis)]

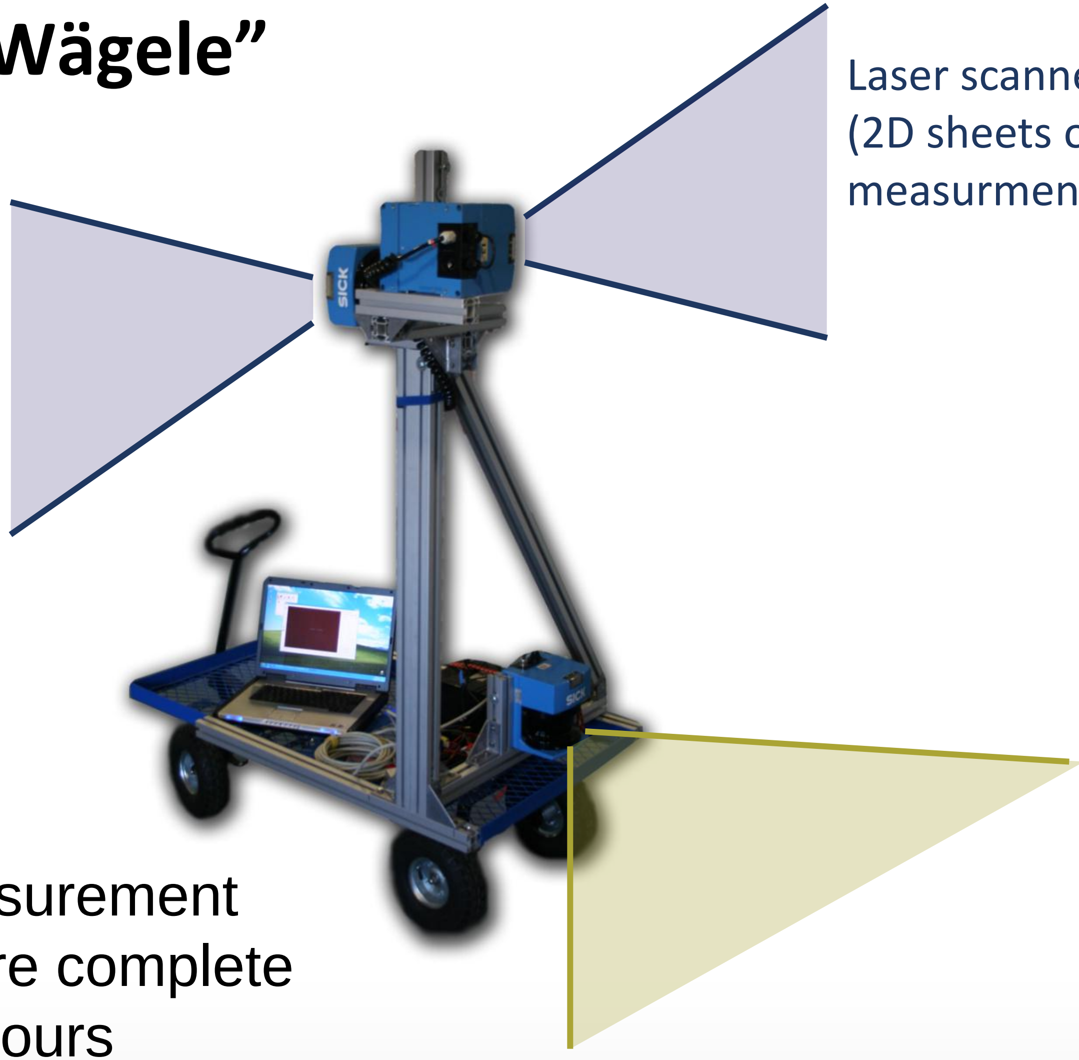
Example

Example: The Stanford “Digital Michelangelo Project”

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

Scanning the World....

Example: The “Wägele”



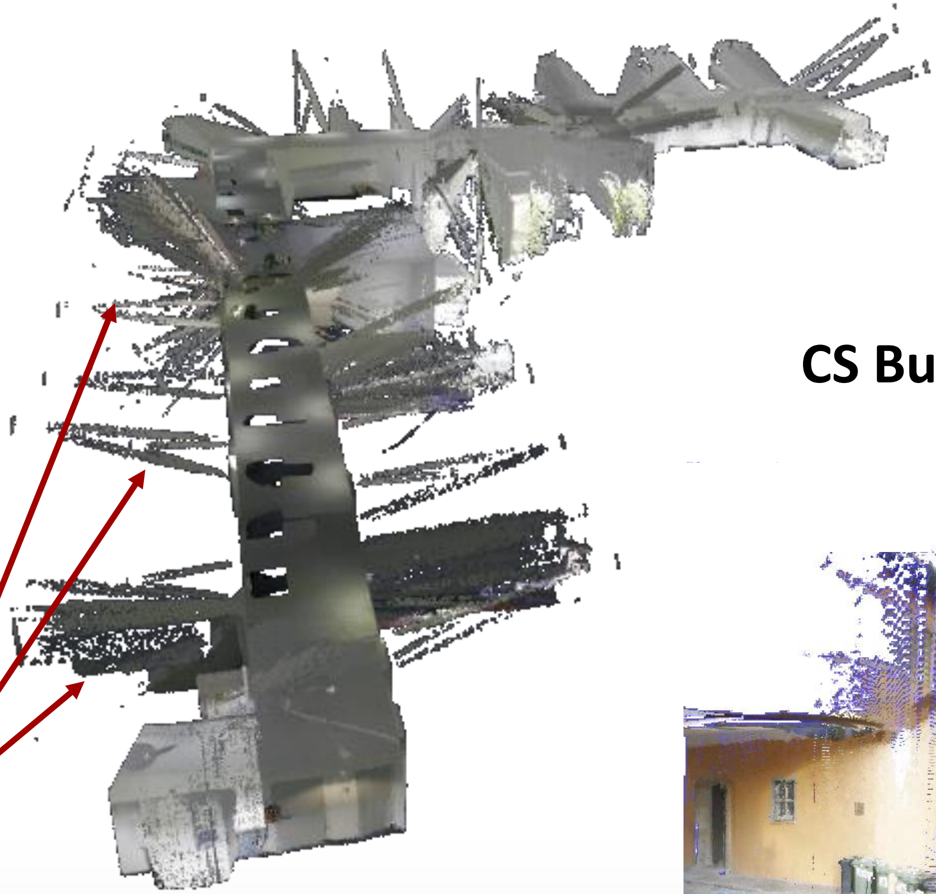
Laser scanners
(2D sheets of distance
measurments)

[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)

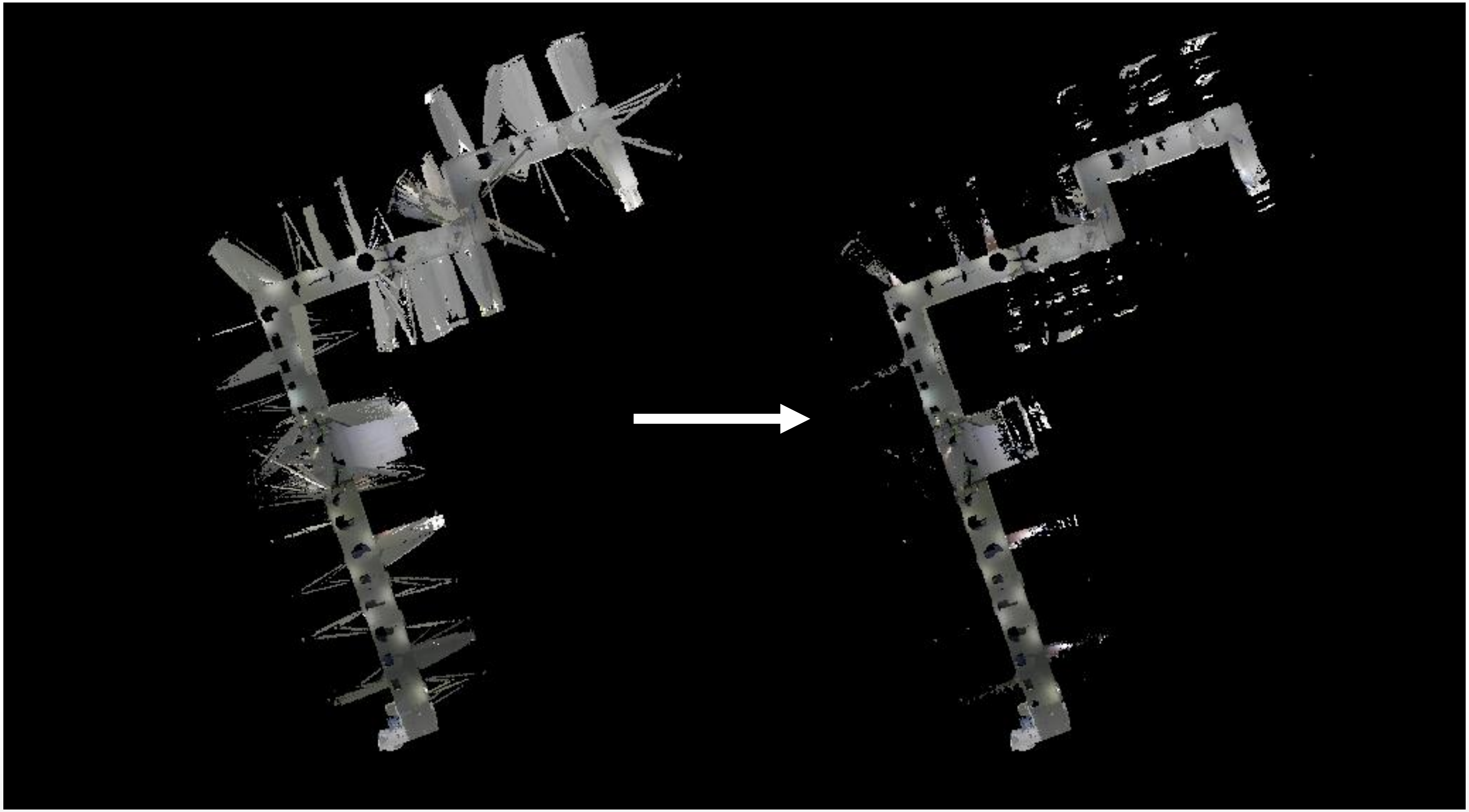


...lots of artifacts
(the scanner does not really like windows)

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)

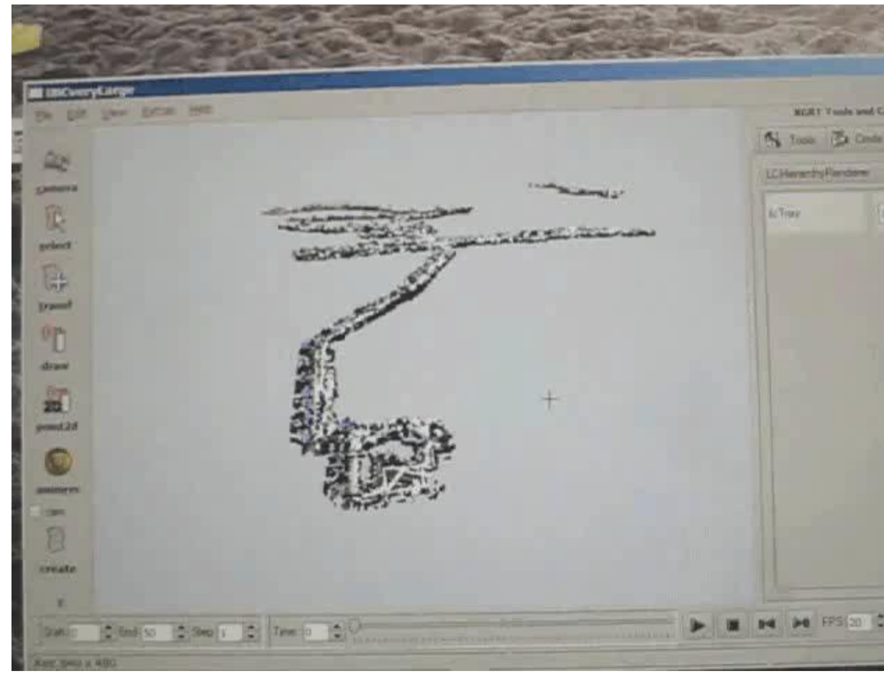
City Scanning

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

[data set: Institute for Cartography, Leibnitz University Hannover]

HUGE Data Sets

The Largest Data Set Currently 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

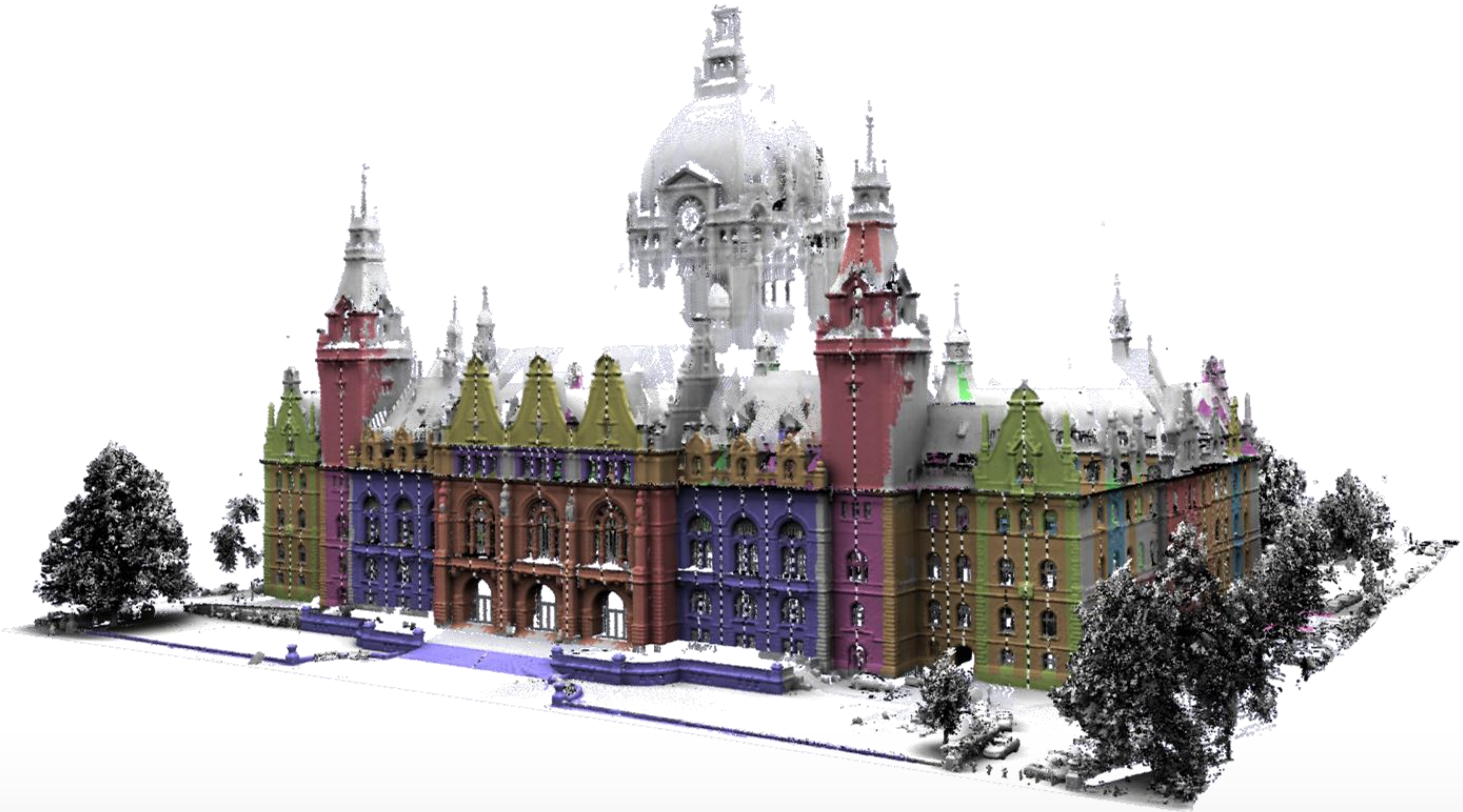
Geometry Processing

Examples of Our Own Research

Symmetry Detection

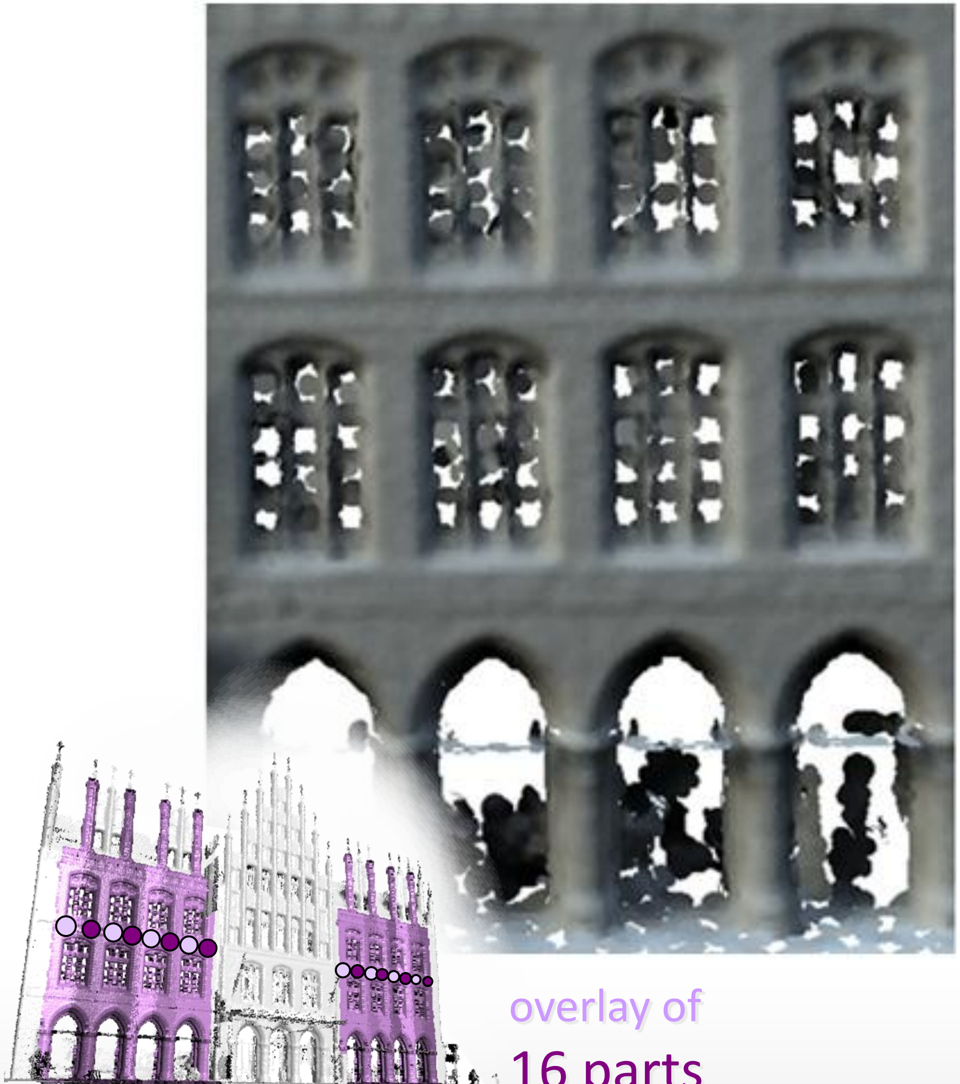
[data sets: C. Brenner, IKG, Universität Hannover]

Symmetry Detection



[data sets: C. Brenner, IKG, Universität Hannover]

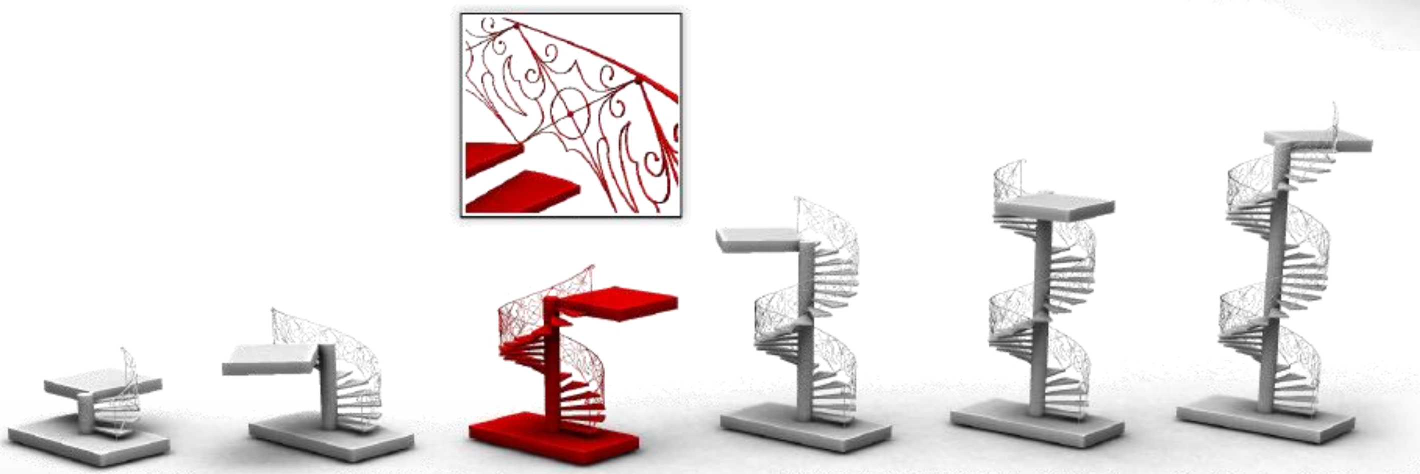
Reconstruction by Symmetry



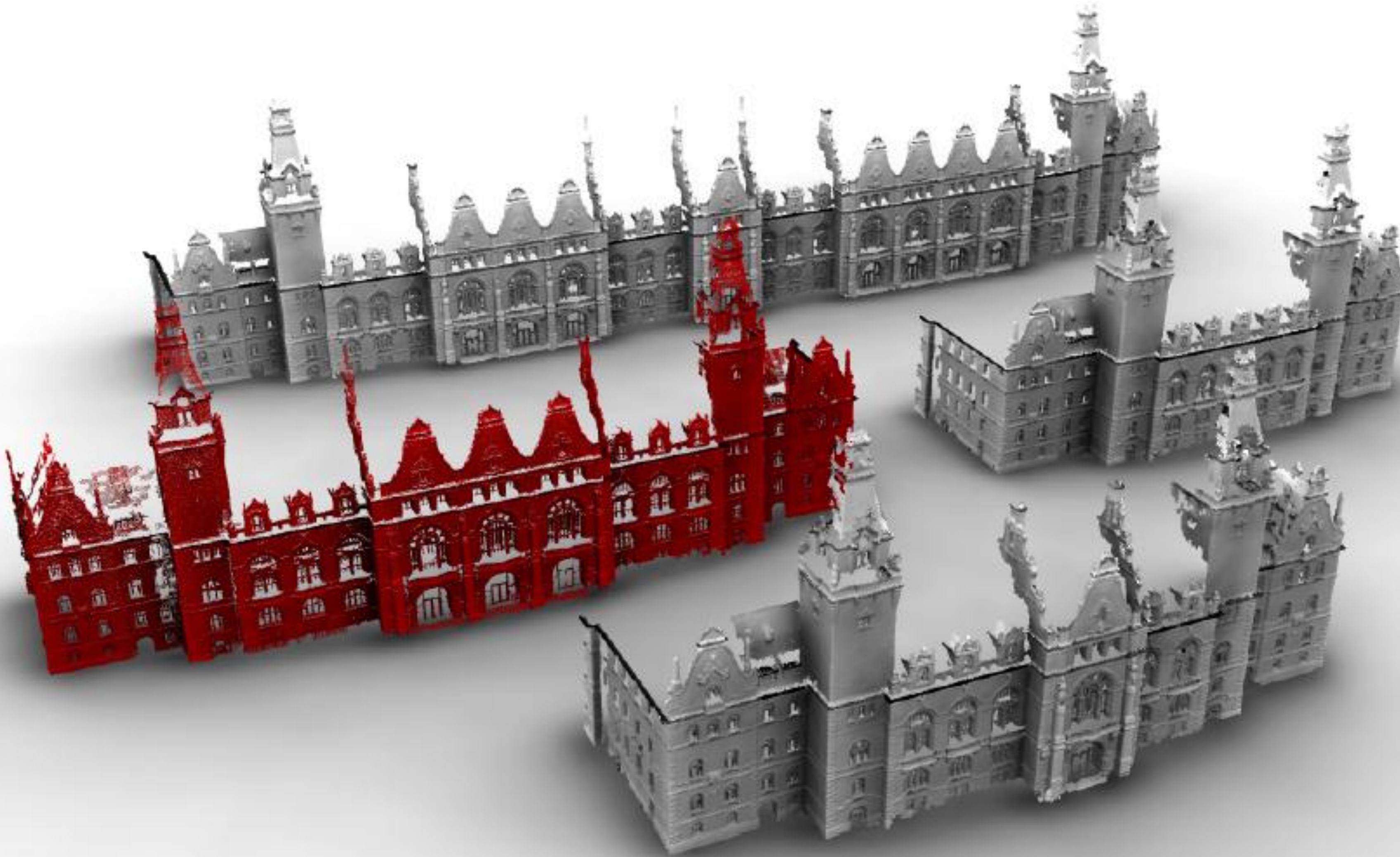
overlay of
16 parts

[data sets: C. Brenner, IKG, Universität Hannover]

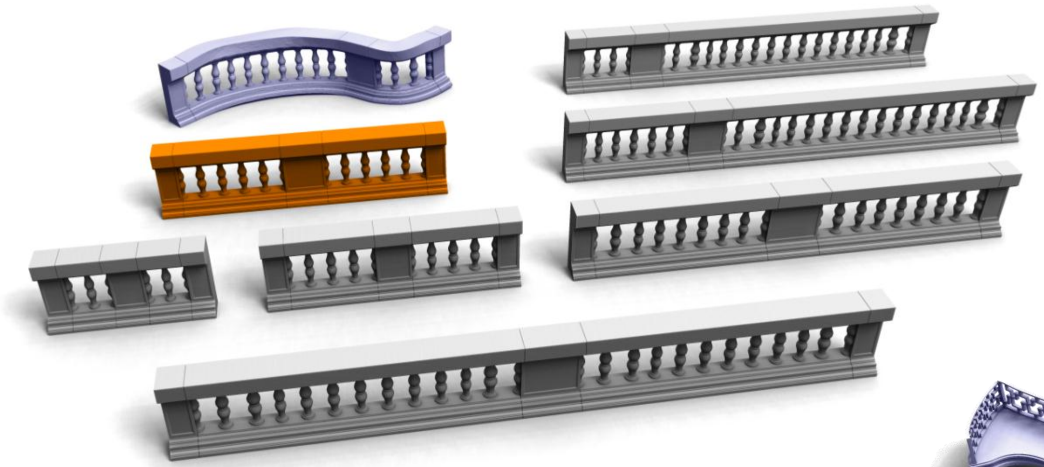
Results

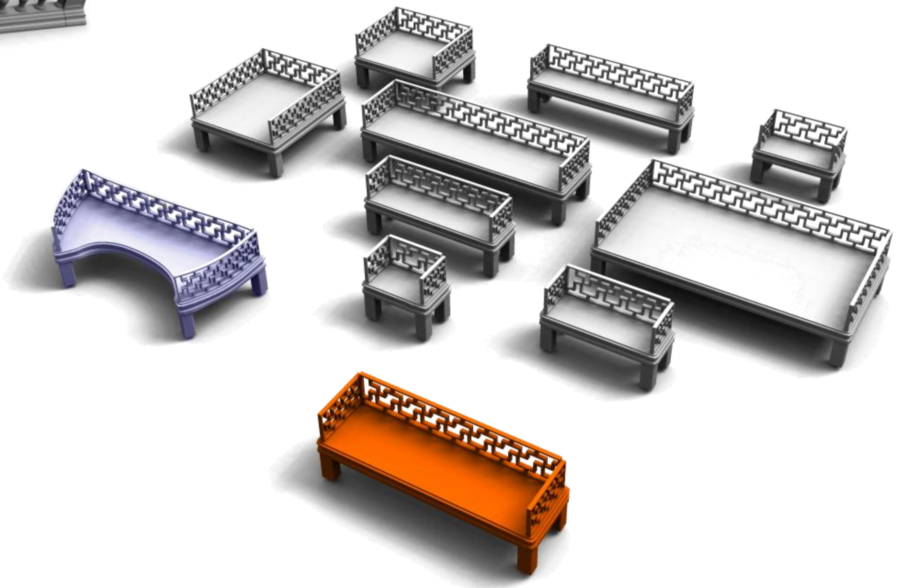


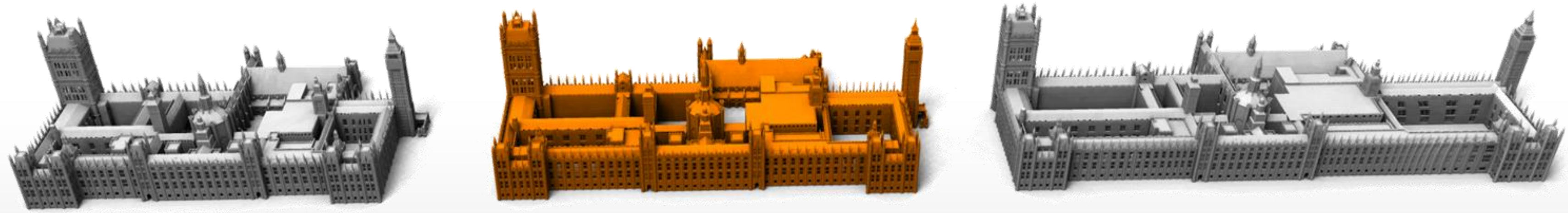
Results



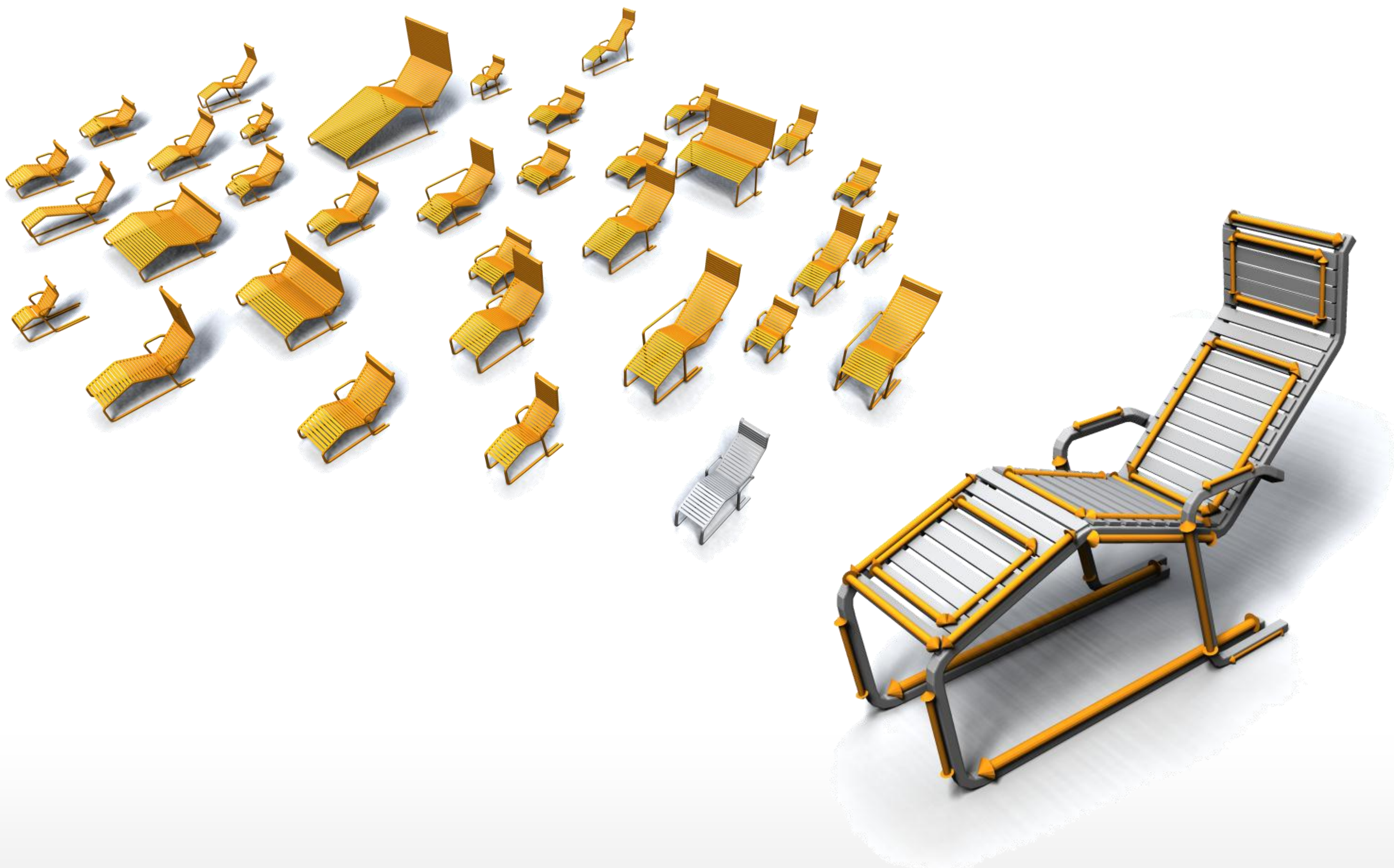
Regularity Aware Deformation







Algebraic Resizing

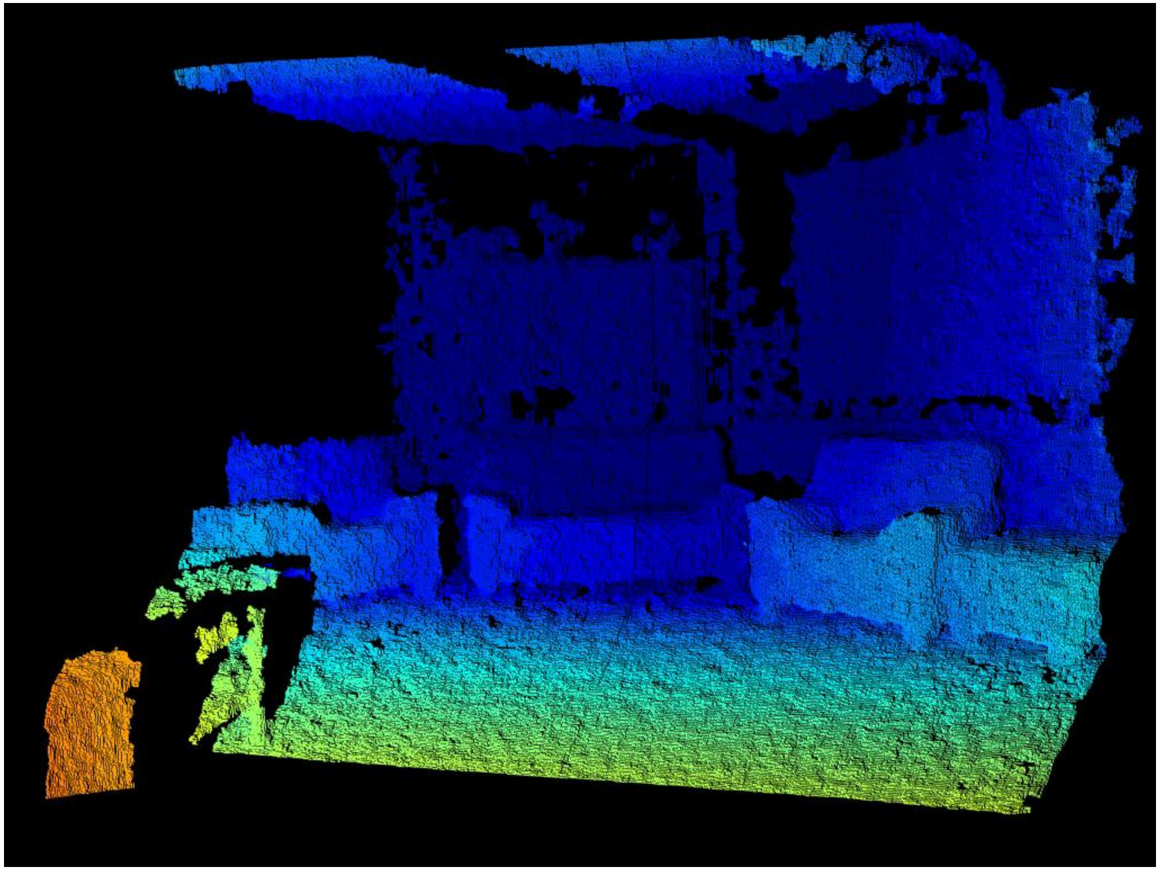


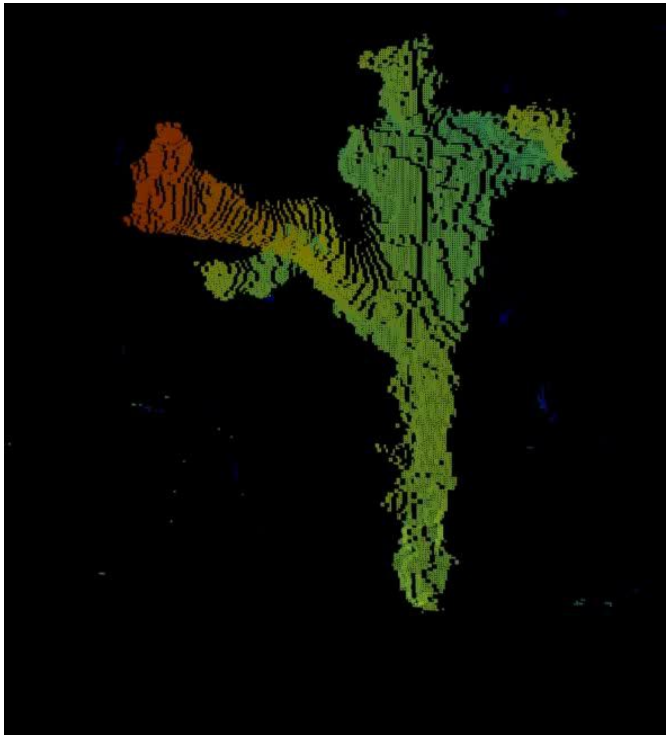
Scanning Moving Geometry

Real-time 3D scanners:

- Acquire geometry at video rates
- Capture 3D movies: “performance capture”
- Not done yet – highly active research area

Kinect Example Data

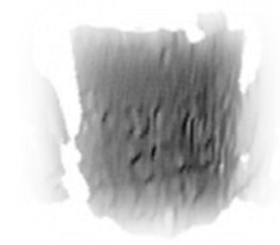




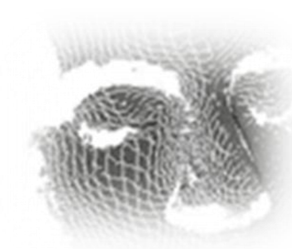
Animation Reconstruction

Problems

- Noisy data
- Incomplete data (acquisition holes)
- No correspondences
- Just point clouds



noise

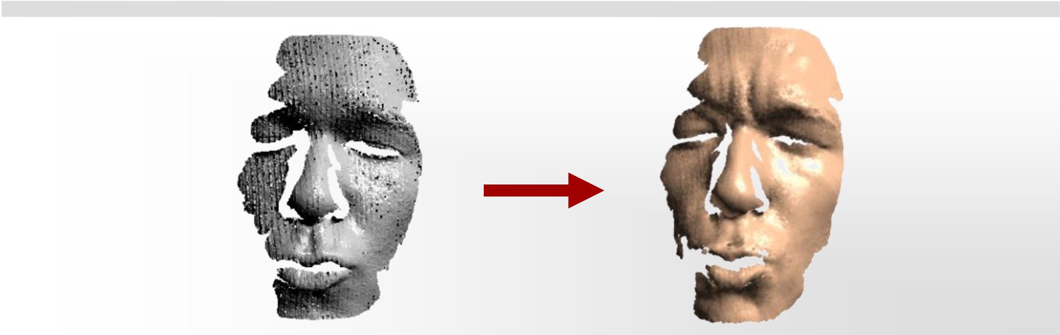


holes

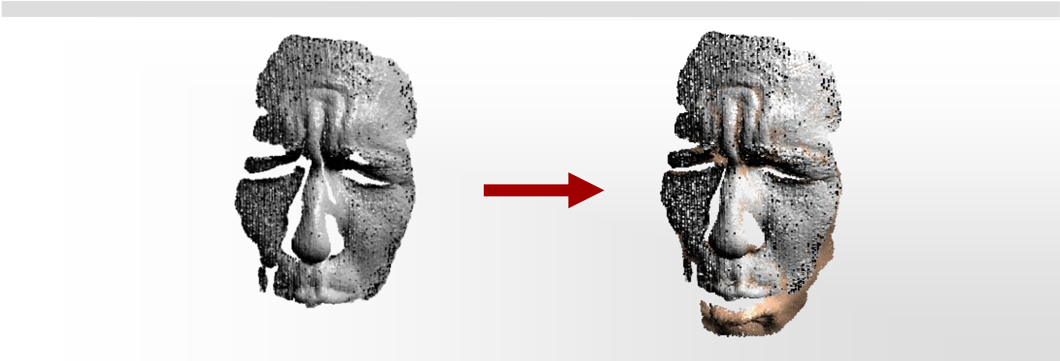
missing correspondences

Animation Reconstruction

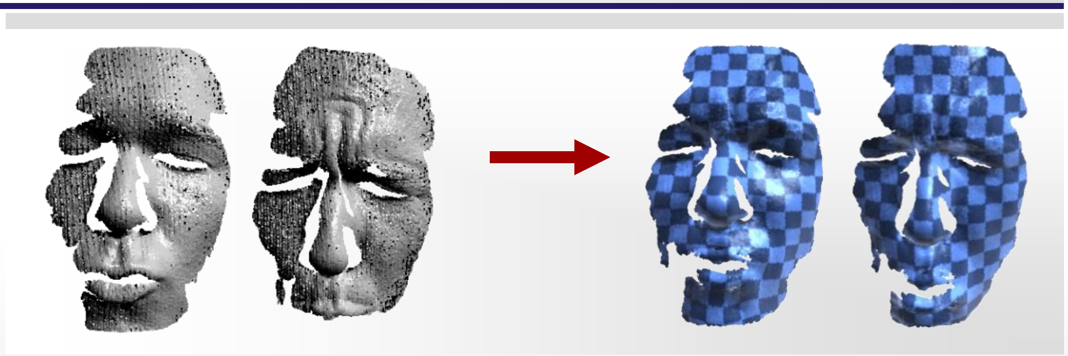
Remove noise, outliers

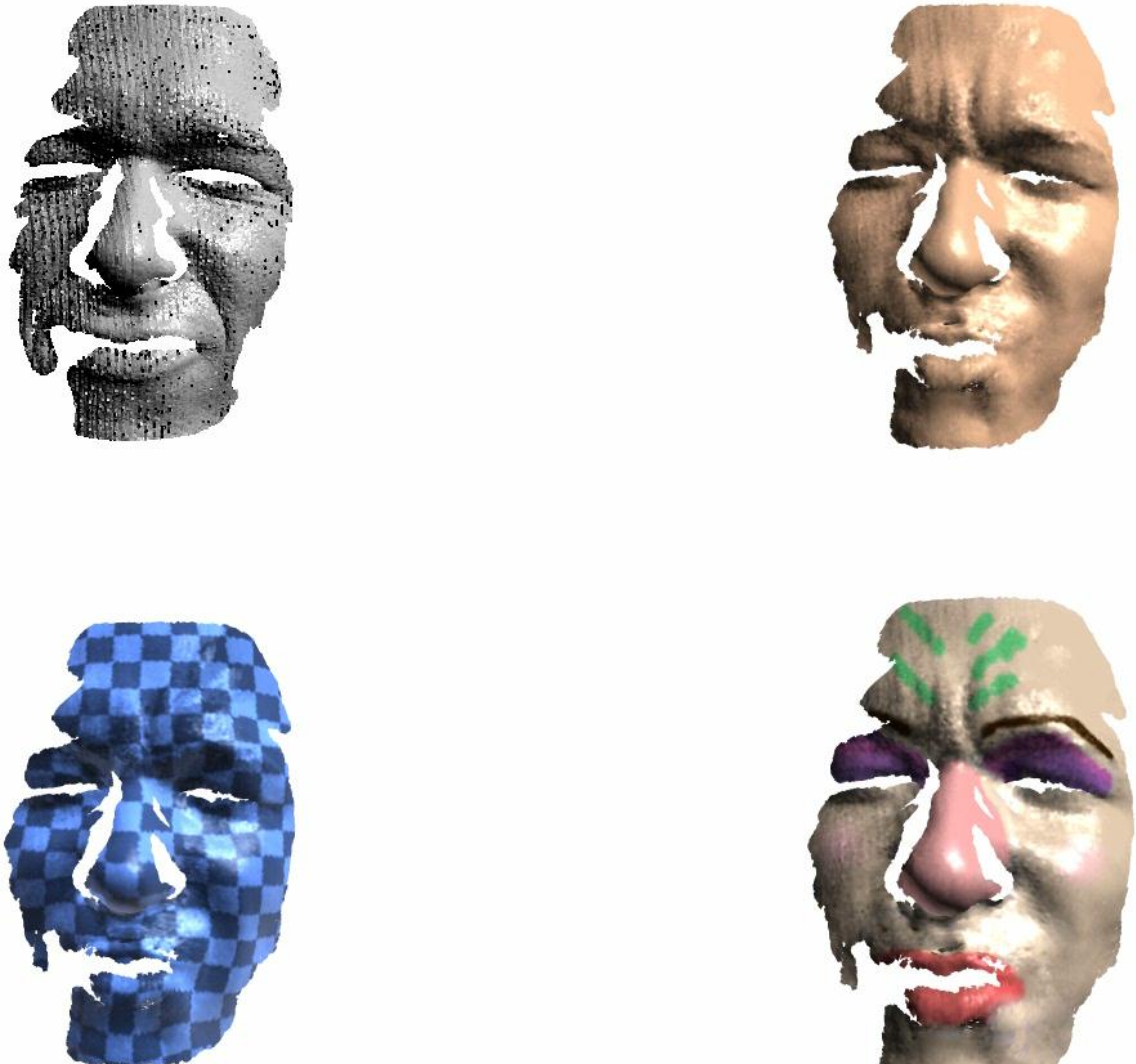


Fill-in holes
(from all frames)

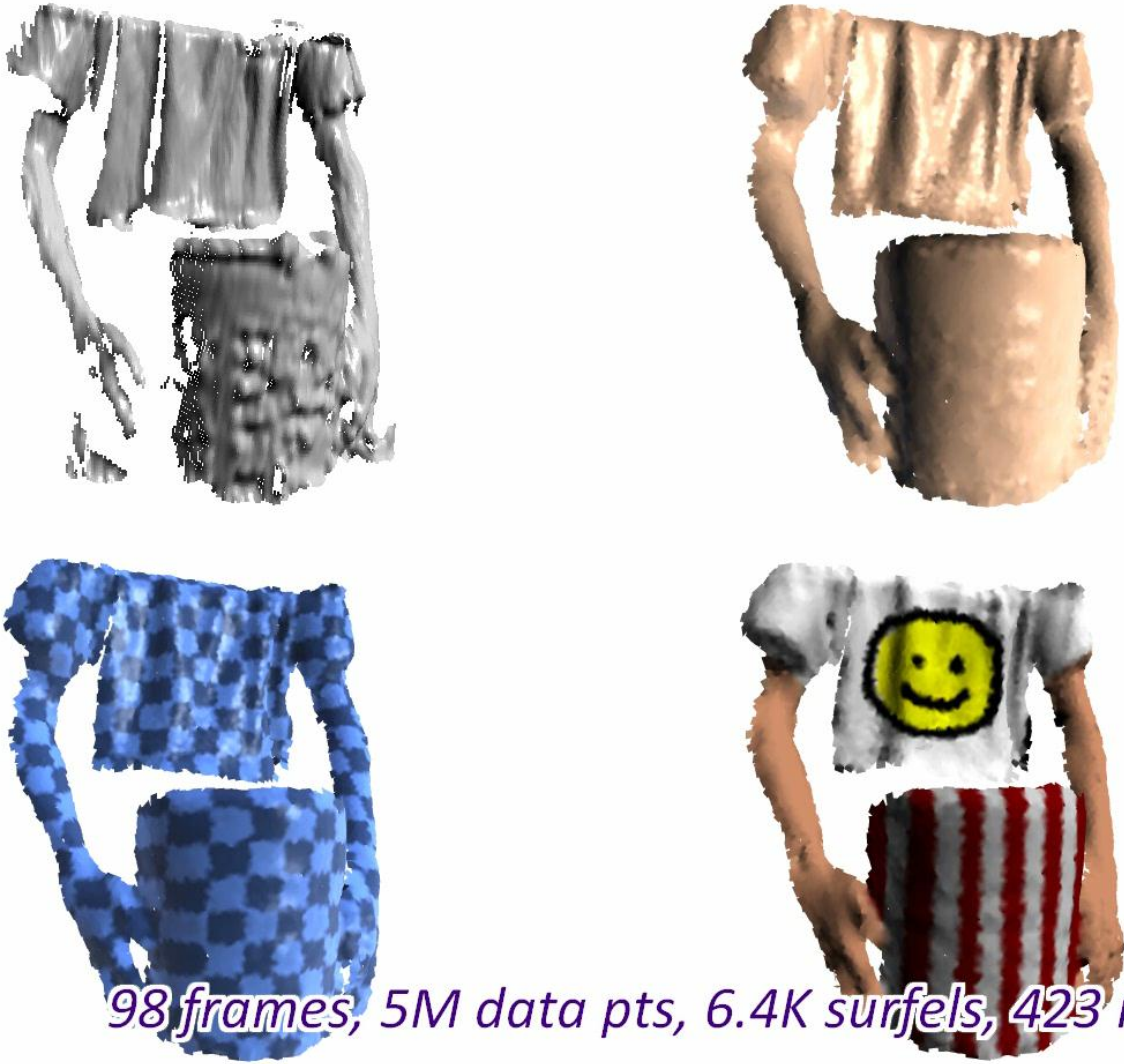


Dense correspondences





79 frames, 24M data pts, 21K surfels, 315 nodes



98 frames, 5M data pts, 6.4K surfels, 423 nodes

Lecture Overview

Topics

Geometric Modeling 2012

Mathematical Background

- Function spaces
- Differential geometry

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

Geometric Modeling 2012

Geometry Processing

- 3D Scanner data
- Registration
- Surface reconstruction
- Point cloud processing

Preliminary List:

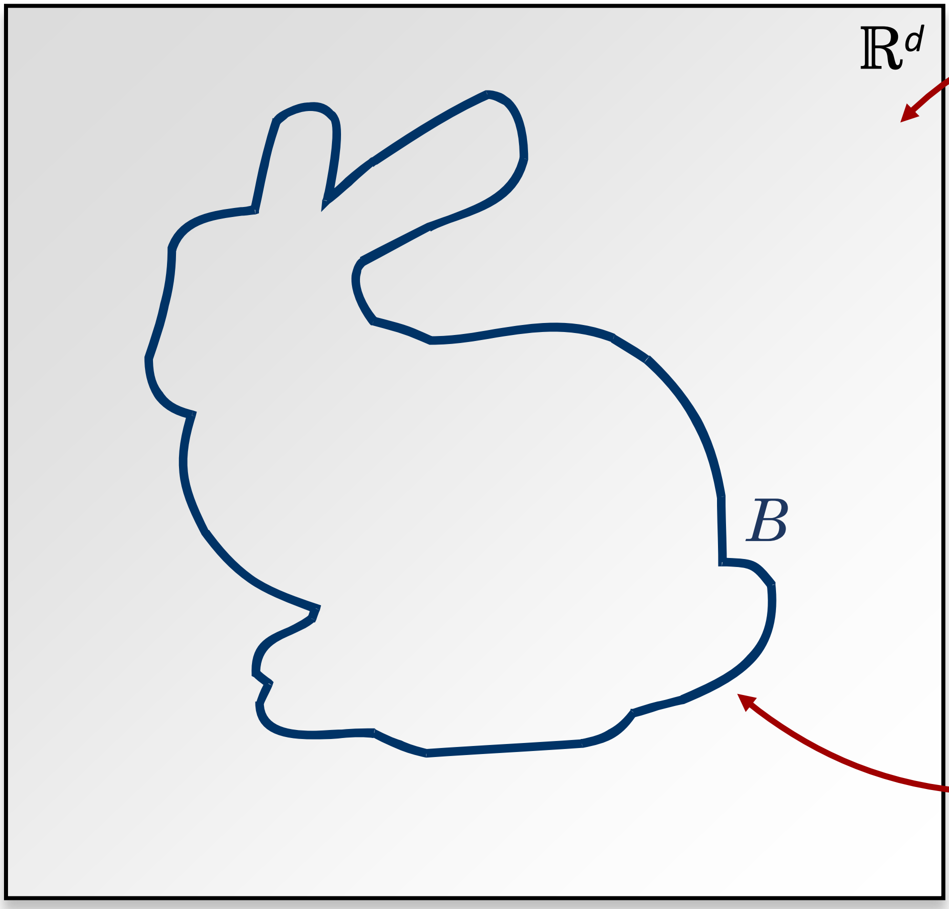
- Subject to change & reordering

Overview

Modeling Techniques

Geometric Modeling

What do we want to do?

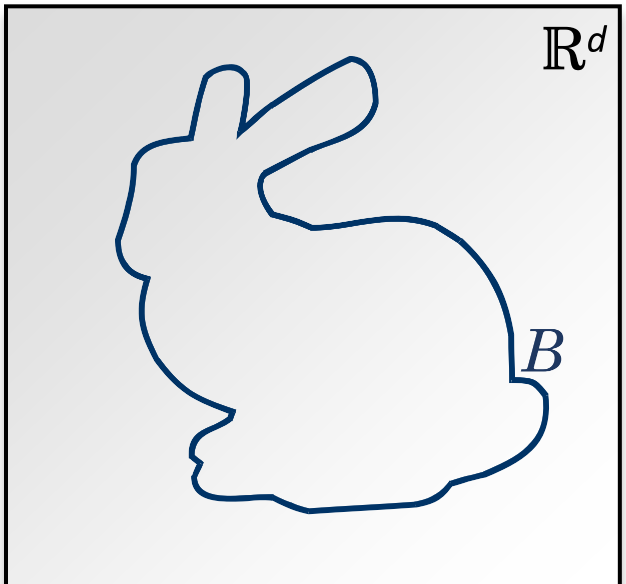


empty space
(typically $\mathbb{R}^3$)

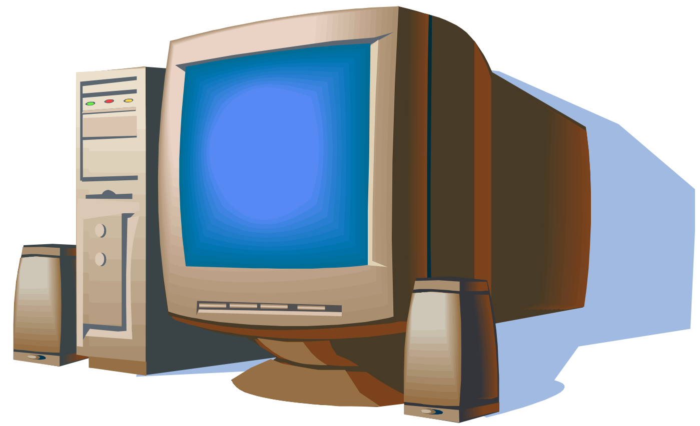
geometric object
$B \subseteq \mathbb{R}^3$

Fundamental Problem

The Problem:



infinite number of points



my computer: 4GB of memory

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

Modeling Approaches

Two Basic Approaches

- Discrete representations
  - “Pixels”
  - Fixed discrete bins
- “Continuous” representations
  - “Vector graphics”
  - Mathematical description
  - Evaluate continuously

Discrete Representations

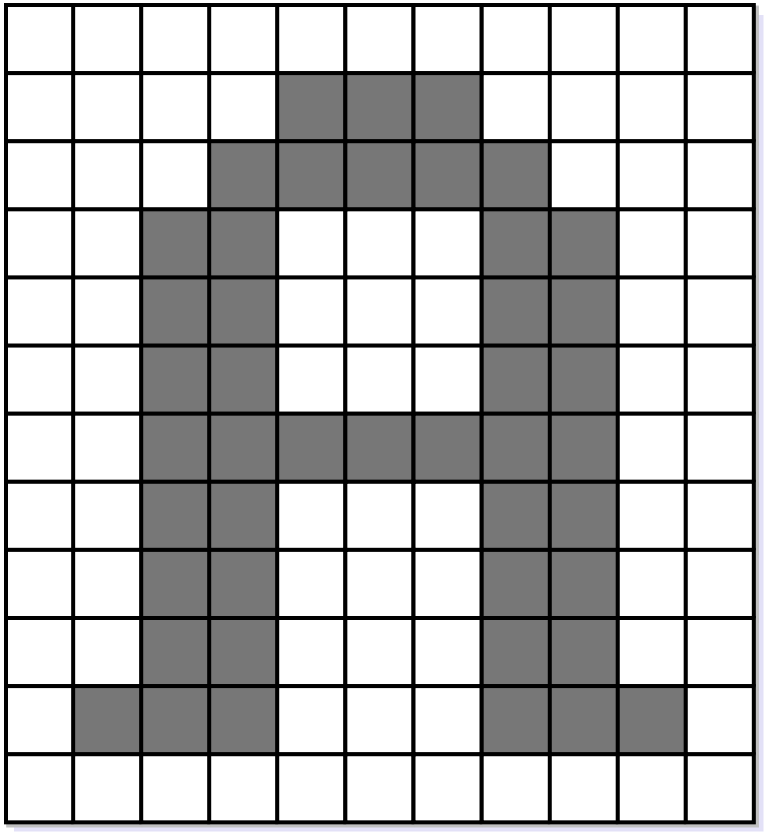
Discrete represenations

- Fixed Grid of values:

  $(i_1, \ldots, i_{d_s}) \in \mathbb{Z}^{ds} \rightarrow (x_1, \ldots, x_{d_t}) \in \mathbb{R}^{dt}$

- Typical scenarios:
  - $d_s = 2$, $d_t = 3$: Bitmap images
  - $d_s = 3$, $d_t = 1$: Volume data
  - $d_s = 2$, $d_t = 1$: Depth maps
- PDEs / Numerics

  "Finite Differences" models



Modeling Approaches

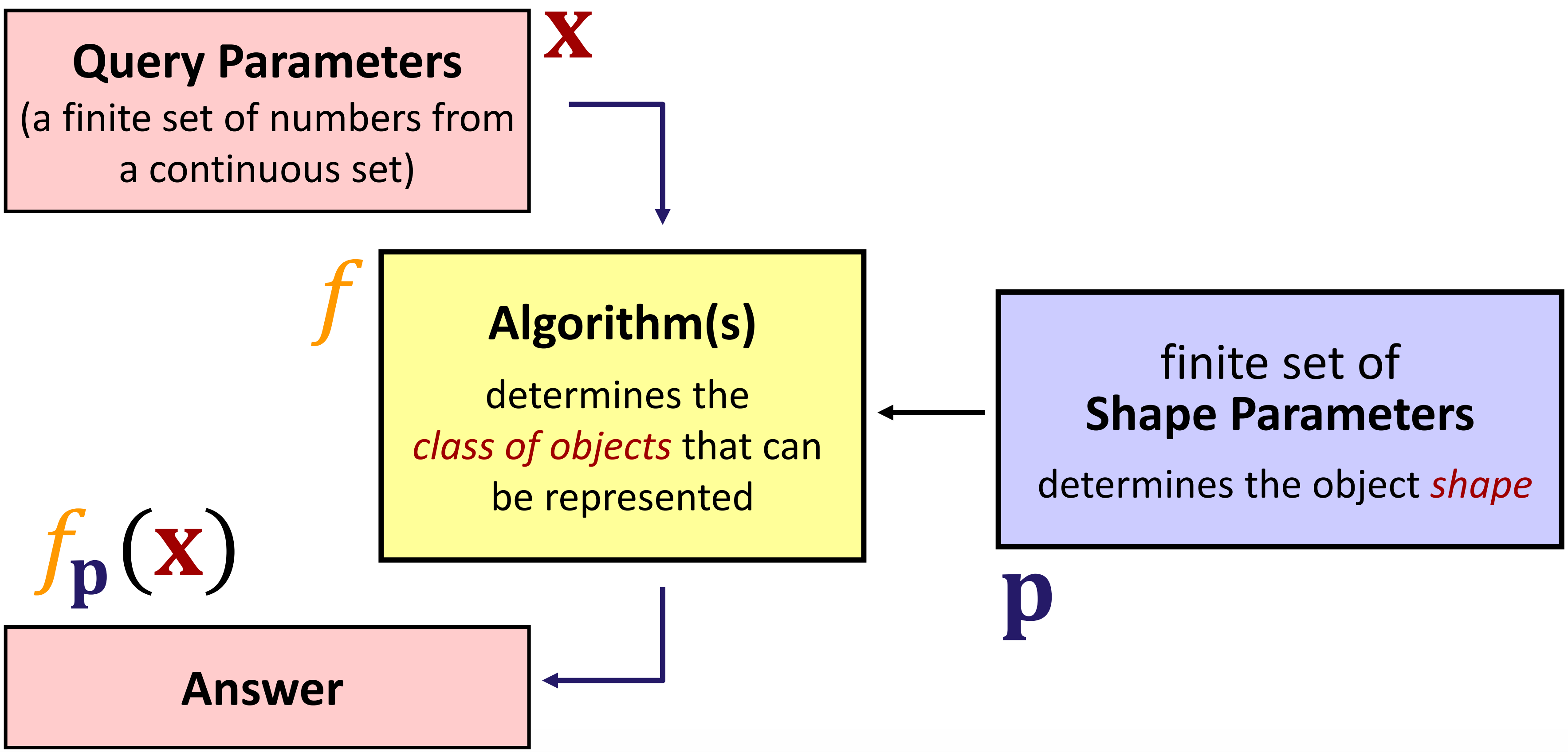
Two Basic Approaches

- Discrete representations
  - "Pixels"
  - Fixed discrete bins
- "Continuous" representations
  - "Vector graphics"
  - Mathematical description
  - Evaluate continuously

Designing Functions

Continuous Models

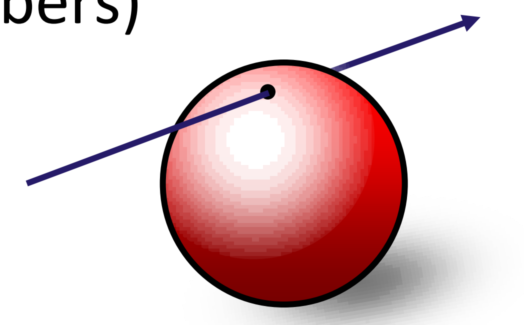
Basic principle: procedural modeling

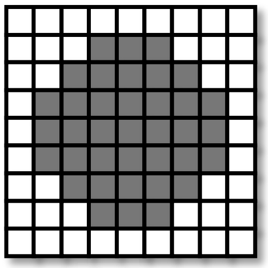


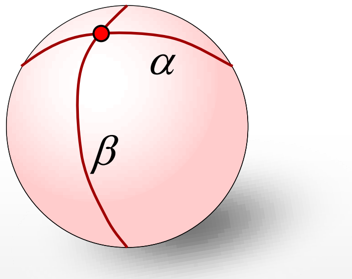
Example: Continuous Model

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 (α, β)
    - Output: position (3 numbers)







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}

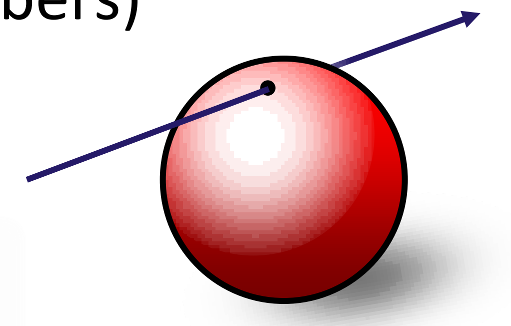
$$f: \mathbb{R}^5 \to [0,1]$$

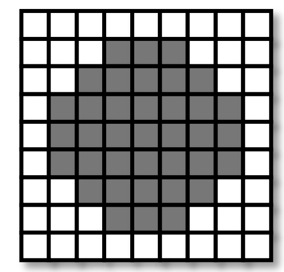
  - **Inside/outside test** (e.g. for rasterization)
    - Input: Position (3 numbers)
    - Output: {true, false}

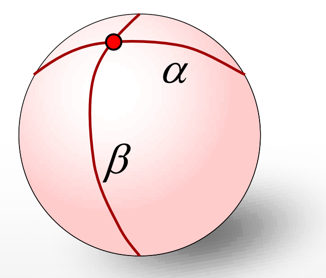
$$f: \mathbb{R}^3 \to [0,1]$$

  - **Parametrization** (e.g. for display)
    - Input: longitude, latitude (α, β)
    - Output: position (3 numbers)

$$f: \mathbb{R}^2 \to \mathbb{R}^3$$







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...

And: algorithms to construct and alter models

Continuous, Procedural Models

"Continuous" representations

- Algorithm (math: *function*) describes the shape
  - Definition: *finite number* of *continuous parameters*
  - Query: *finite number* of *continuous parameters*
- Characteristics:
  - More involved (have to ask for information)
  - Potentially "infinite" resolution (continuous model)
- Structural complexity limited by algorithm

This lecture: focus on these represenations

- Mathematically, we study function design
- Mostly linear design approaches
  - Just find a basis to a linear vector space, that's all the magic

Classes of Models

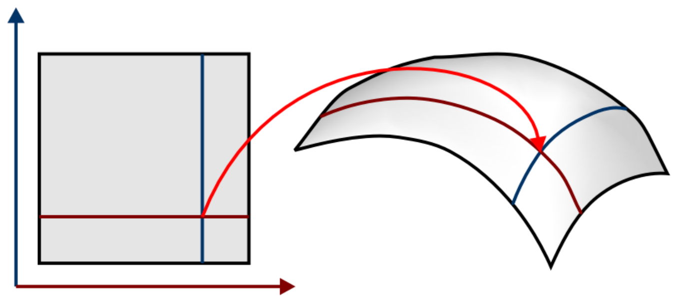
(Main) classes of models in this lecture:

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

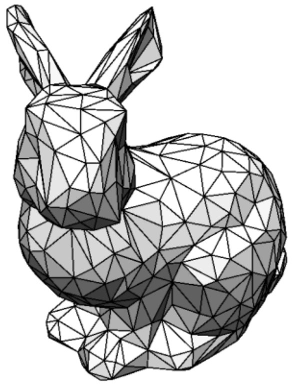
Remarks

- 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

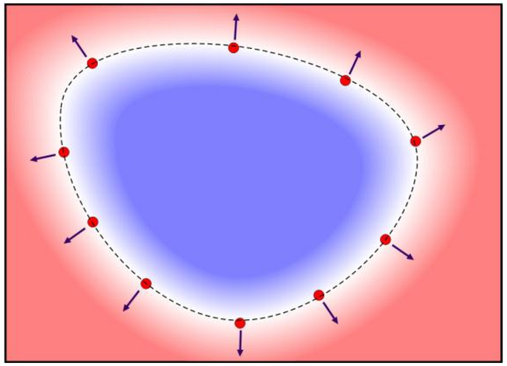
Modeling Zoo



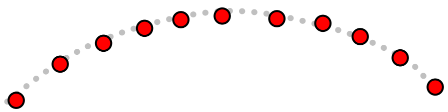
Parametric Models



Primitive Meshes

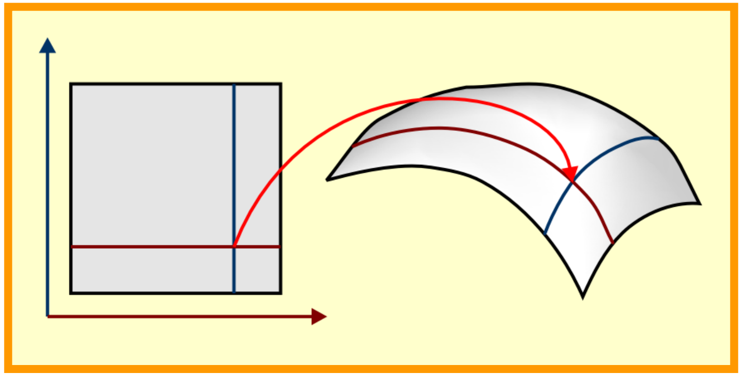


Implicit Models

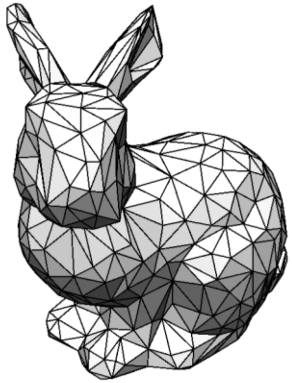


Particle Models

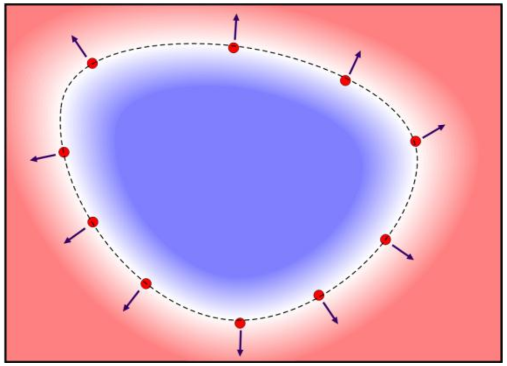
Modeling Zoo



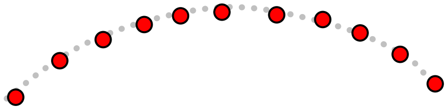
Parametric Models



Primitive Meshes

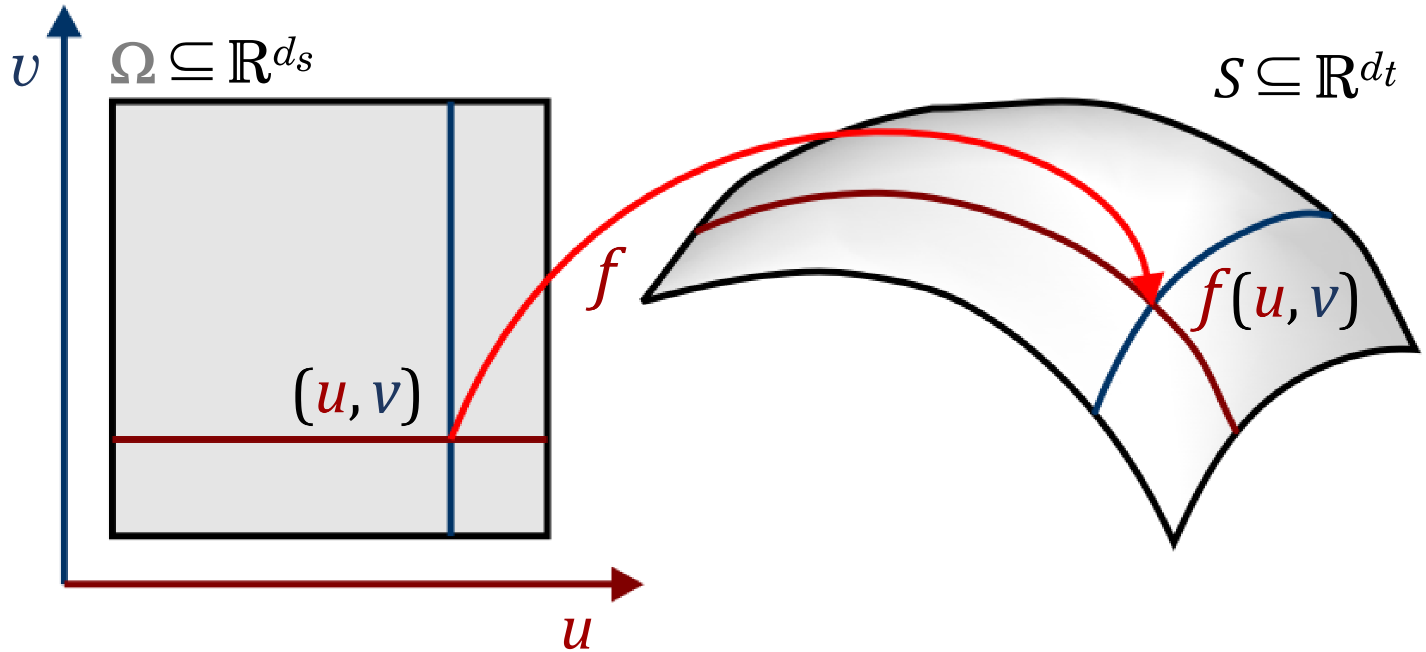


Implicit Models



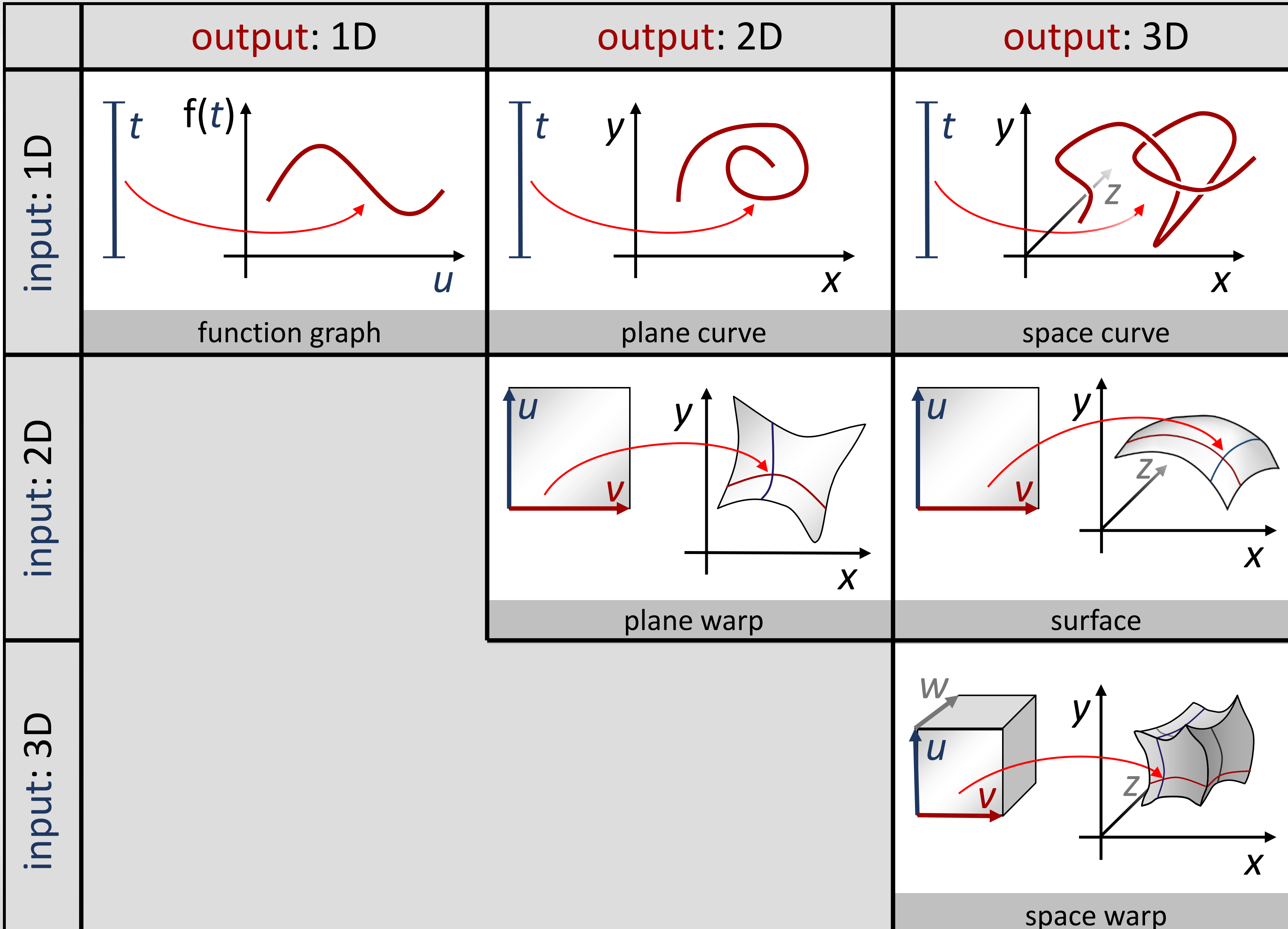
Particle Models

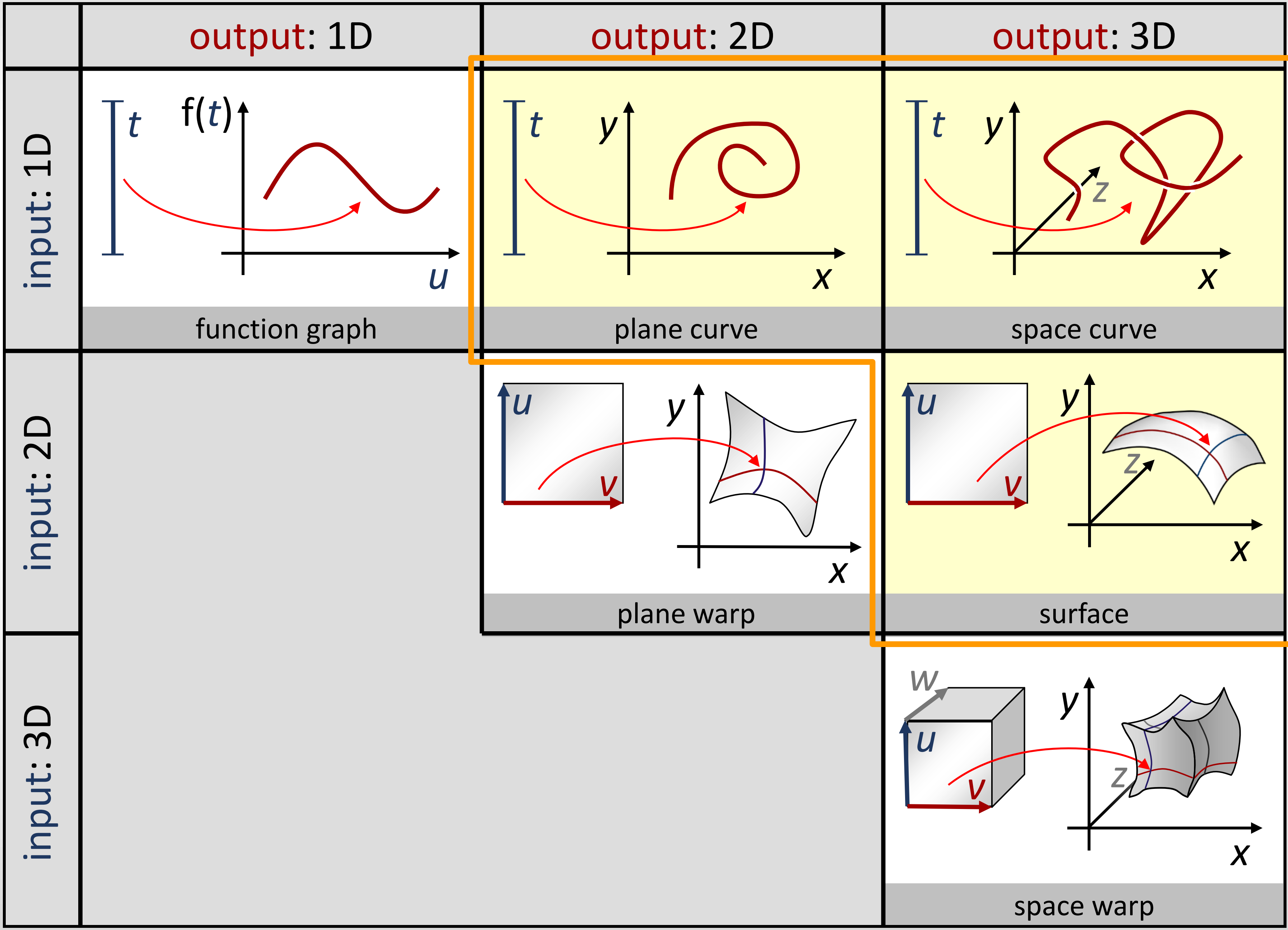
Parametric Models



Parametric Models

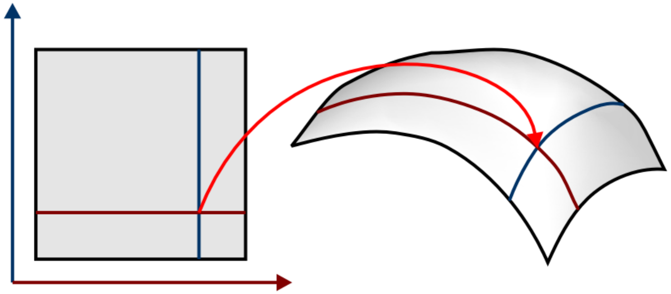
- Function f maps from parameter domain Ω to target space
- Evaluation of f gives one point on the model


output: 1D
output: 2D
output: 3D
input: 1D
input: 2D
input: 3D
t
f(t)
u
function graph
t
y
x
plane curve
t
y
z
x
space curve
u
v
y
x
plane warp
u
v
y
z
x
surface
w
u
v
y
z
x
space warp

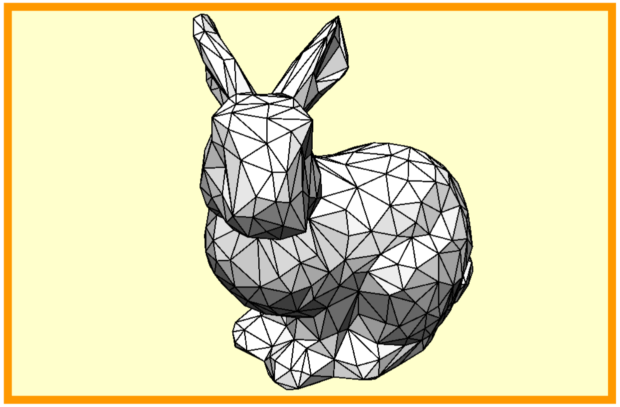


output: 1D
output: 2D
output: 3D
input: 1D
input: 2D
input: 3D
function graph
plane curve
space curve
plane warp
surface
space warp

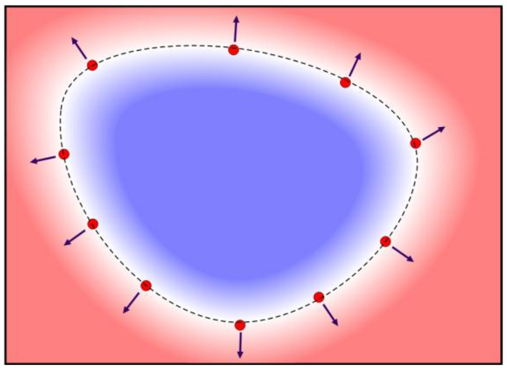
Modeling Zoo



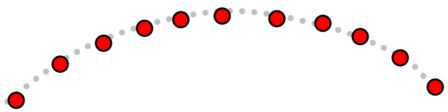
Parametric Models



Primitive Meshes



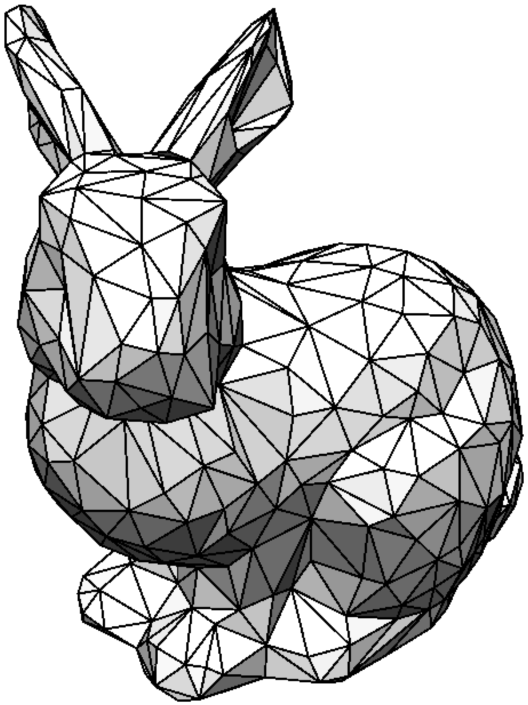
Implicit Models



Particle Models

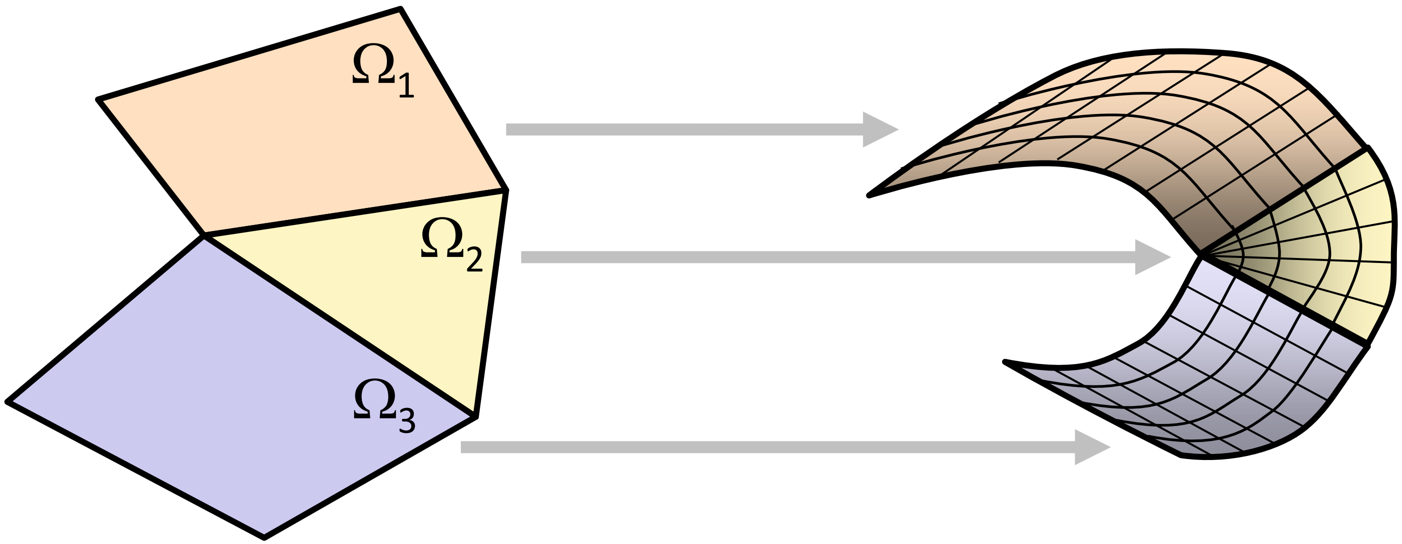
Primitive Meshes

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”)

Primitive Meshes



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:

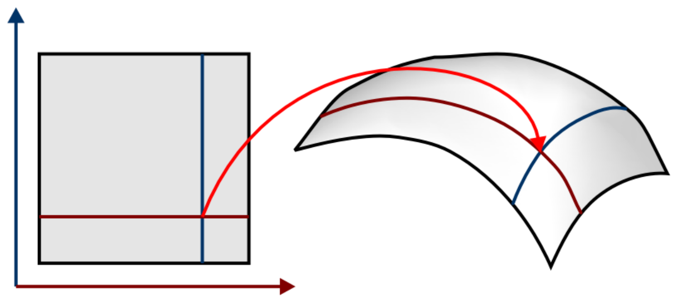
- Compact representation (usually)
- Can represent arbitrary topology
- Important objects can be represented exactly
  - When using the right parametric parts
  - E.g. NURBS: circles, cylinders, spheres $\rightarrow$ CAD/CAM

Meshes are not so great

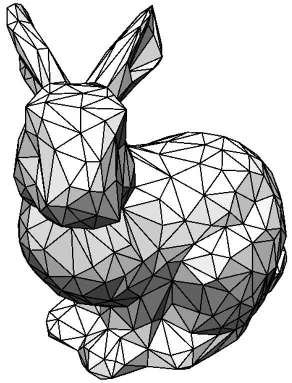
Problem with Meshes:

- 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
    $\Rightarrow$ Changing object topology is painful
- Sometimes difficult to use
  - Rule of thumb:
    drastic topology changes $\Rightarrow$ meshes are hard to use
  - Extreme example: fluid simulation (splashing water)

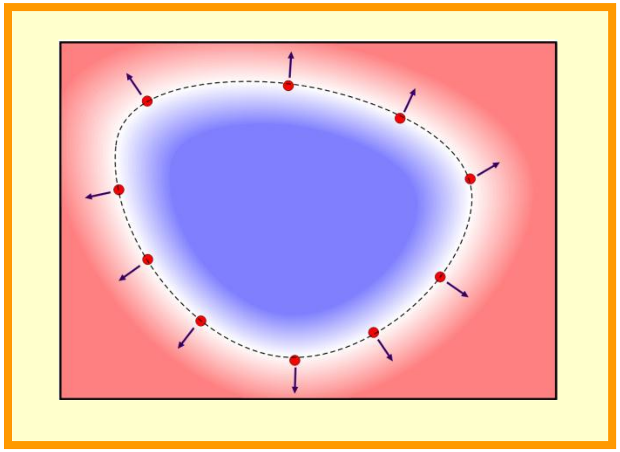
Modeling Zoo



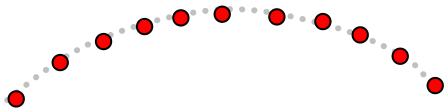
Parametric Models



Primitive Meshes



Implicit Models



Particle Models

Implicit Modeling

General Formulation:

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

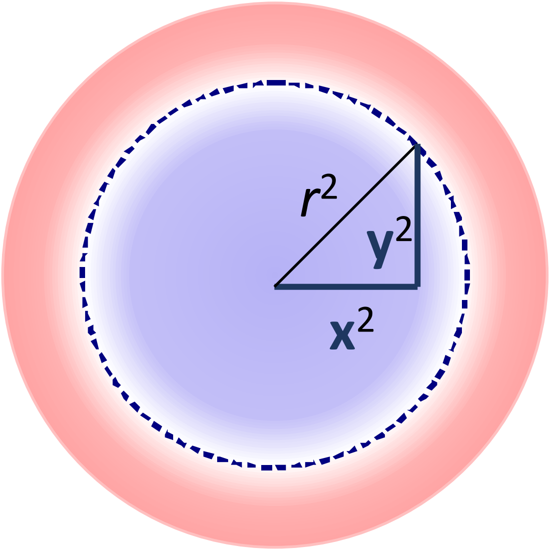
This means...:

- Surface = set of points where f vanishes ($f(\mathbf{x}) = 0$)
- Also known as "*level-set methods*"
- Alternative notation: $S = f^{-1}(0)$

Implicit Modeling

Example:

- Circle: $\mathbf{x}^2 + \mathbf{y}^2 = r^2$
  $\Leftrightarrow f_r(\mathbf{x},\mathbf{y}) = \mathbf{x}^2 + \mathbf{y}^2 - r^2 = 0$
- Sphere: $\mathbf{x}^2 + \mathbf{y}^2 + \mathbf{z}^2 = r^2$



Special Case:

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

Implicit Modeling: Pros & Cons

Advantages:

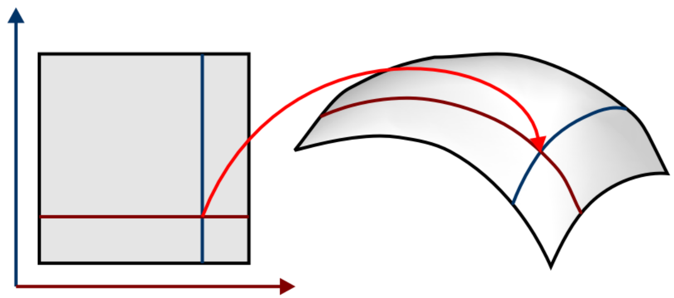
- More general than parametric techniques
- Topology can be changed easily
- Standard technique for simulations with *free boundaries*.
  - Example: Fluid simulation (evolving water-air interface)
  - Geometric modeling: Surface reconstruction, “blobby surfaces”

Implicit Modeling: Pros & Cons

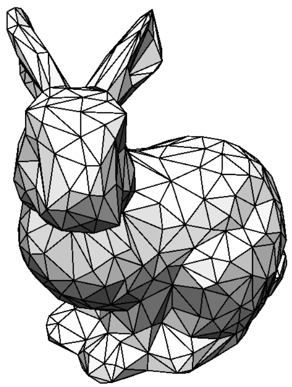
Disadvantages:

- Need to solve inversion: $S = f^{-1}(0)$
- Many algorithms more difficult
  - Difficult: display, surface sampling, conversion (larger run-time costs as well)
  - Easy: inside/outside, Boolean operations
- Memory: Often more costly
  - In particular: sharp boundaries, flat surfaces

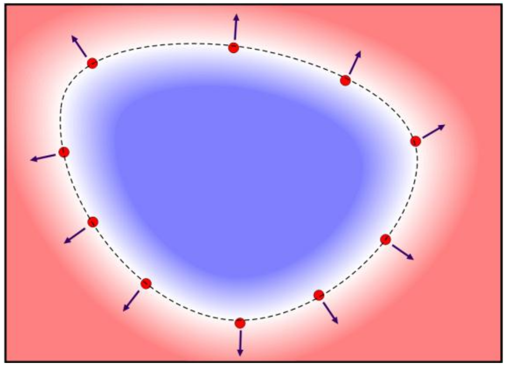
Modeling Zoo



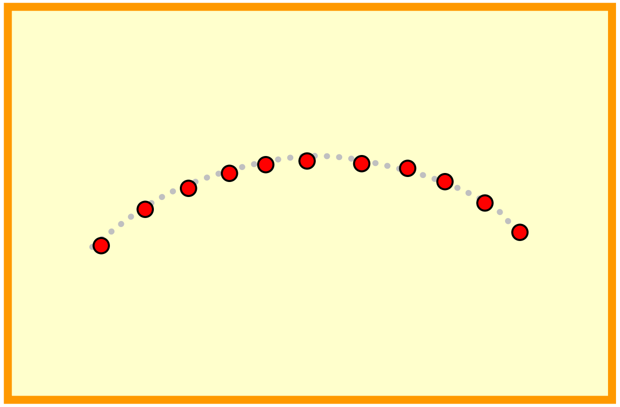
Parametric Models



Primitive Meshes



Implicit Models

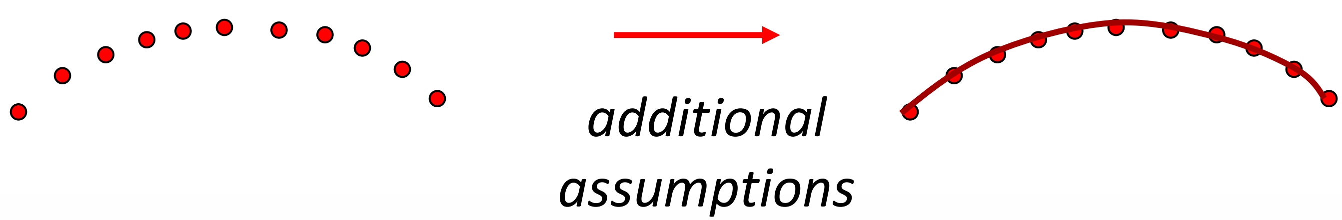


Particle Models

Particle Representations

Particle / Point-based Representations

- Geometry is represented as a set of points / particles
- Irregular *sample* of geometry
- Need additional information to deal with "the empty space around the particles"
  - Reconstruction for processing
  - Also a type of implicit representation



Particle Representations

Helpful Information

- Each particle may carries a set of attributes
  - Must have: position
  - Additional:
    - Particle density (sample spacing)
    - Surface normals
    - Color
    - physical quantities (mass, pressure, temperature), ...
- Additional information might facilitate *reconstruction*

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 scanners yield point clouds
- Have to deal with the problem anyway
- Need algorithms to directly work on “point clouds”
  - (this is the geometry name for particle system)

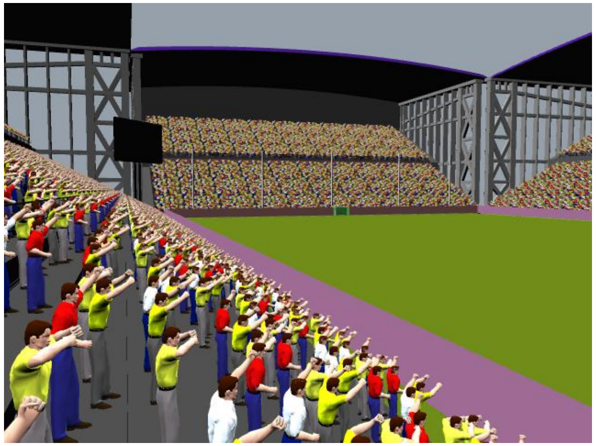
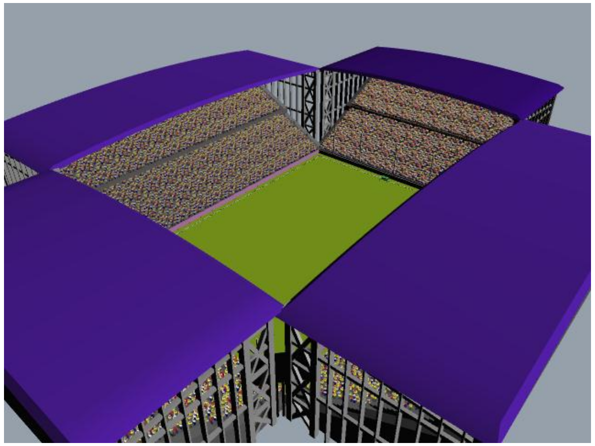
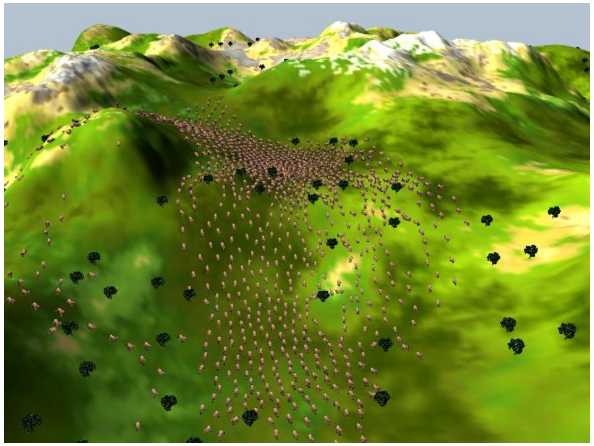
Geometric Modeling

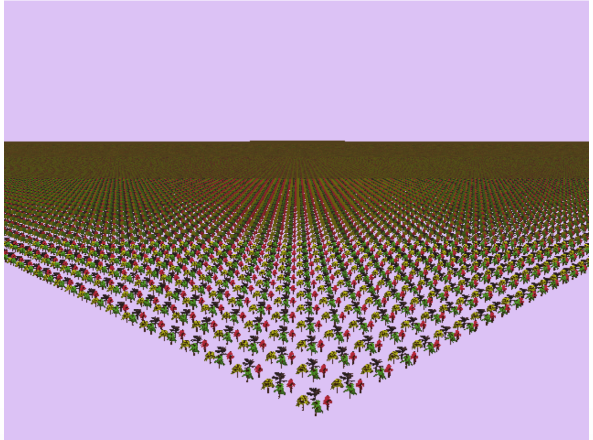
How became the geometric modeling crowd interested in this?

Other Reasons:

- Similar advantages as implicit techniques
- Topology does not matter (for the good and for the bad)
  - Topology is easy to change
  - Topology might be hard to determine
  - 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

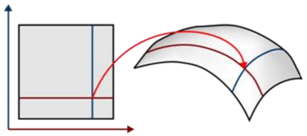






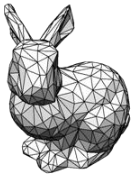

Summary

Summary

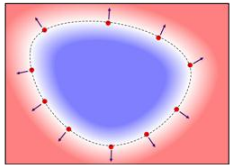
- Lots of different representations
- No silver bullet
- In theory, everything always works, but might be just too complicated/expensive
- Best choice depends on the application
- We will look on all of this...
  - Focus on parametric techniques though
  - Most common approach



Parametric Models



Primitive Meshes



Implicit Models

Particle Models