Voxels, Point Clouds, and Registration

Richard (Hao) Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics

Lecture 3

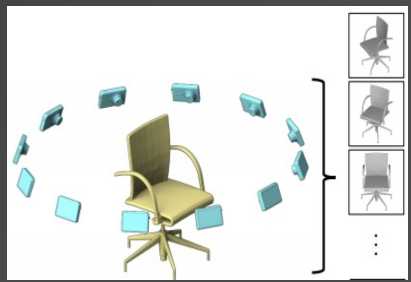
Outline on 3D representations

- Implicit reps
- Parametric reps
- Meshes (subdivision)
- Point clouds
- Volumes
- Projective reps
- Structured reps

Smooth curves and surfaces

Discrete representations

3D → 2D



Parts + relations = structures
Encompasses all low-level reps

Today

- Implicit reps
- Parametric reps

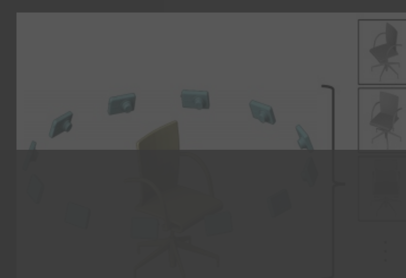
Smooth curves and surfaces

- Meshes (subdivision)
- Point clouds
- Volumes

Discrete representations

- Projective reps

3D → 2D

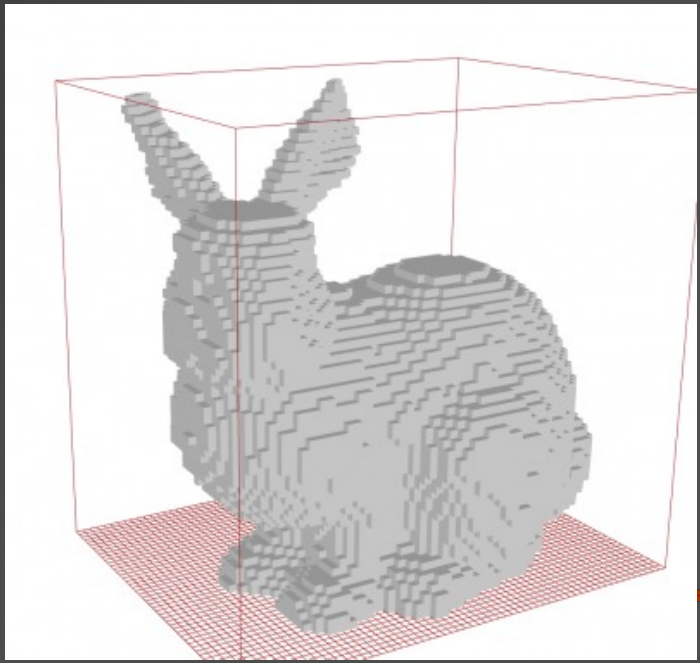


- Structured reps

Parts + relations = structures
Encompasses all low-level reps

Volumetric or voxel representations

- Embed 3D shape in regular volumetric grid: 3D shape = set of all voxels that lie **on or inside** shape
- Closely related to image and pixel representations: it is a **3D image**
- Closely tied to **implicit representations** and support similar operations
- Natural **"first choice" for neuralization** due to similarity to image/pixels

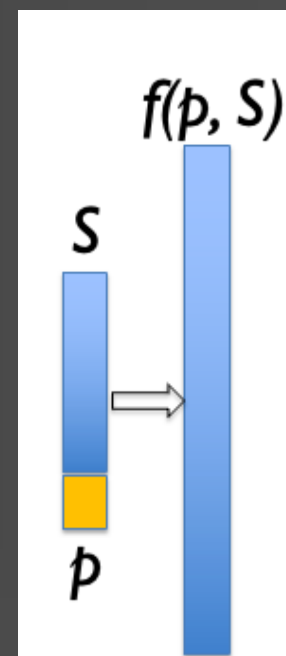


Voxels vs. implicit functions

- Voxels: intrinsically **discrete** representation, limited by resolutions
- Implicit functions: intrinsically **continuous** representation
  - There is an implicit field value for any (x, y, z)
  - When processing an implicit function (e.g., rendering), need to discretize

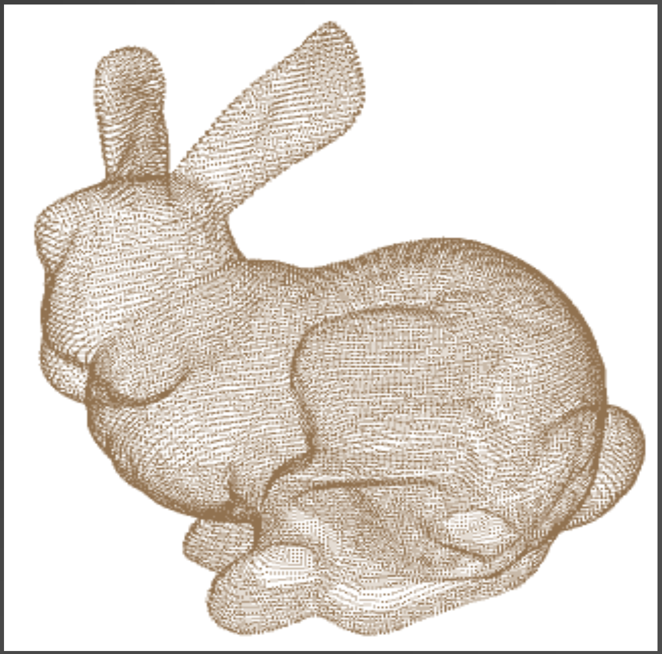
Voxels vs. implicit functions

- Voxels: intrinsically **discrete** representation, limited by resolutions
- Implicit functions: intrinsically **continuous** representation
  - There is a field value for any (x, y, z)
  - When processing an implicit function (e.g., rendering), need to discretize
- IM-Net (OCC-Net, DeepSDF) trained on voxel inputs, e.g., on 64^3 voxels, can learn **continuous** outputs, for all $p \in R^3$



Point-based representation (PBR)

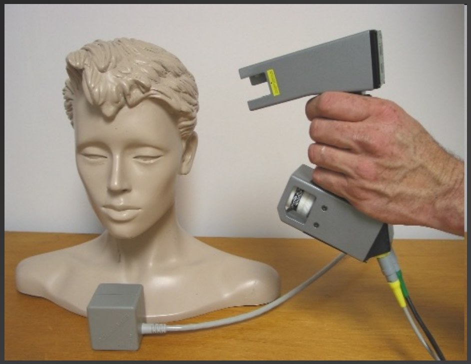
- A 3D surface model is represented using a set of **points near the surface**
- There is no (explicit) connectivity information between the points
- Typically need *k*NN – ***k* nearest neighbors** – during processing
- Point normals can also be specified or estimated for rendering



Point cloud acquisition

Cyberware

InSpeck



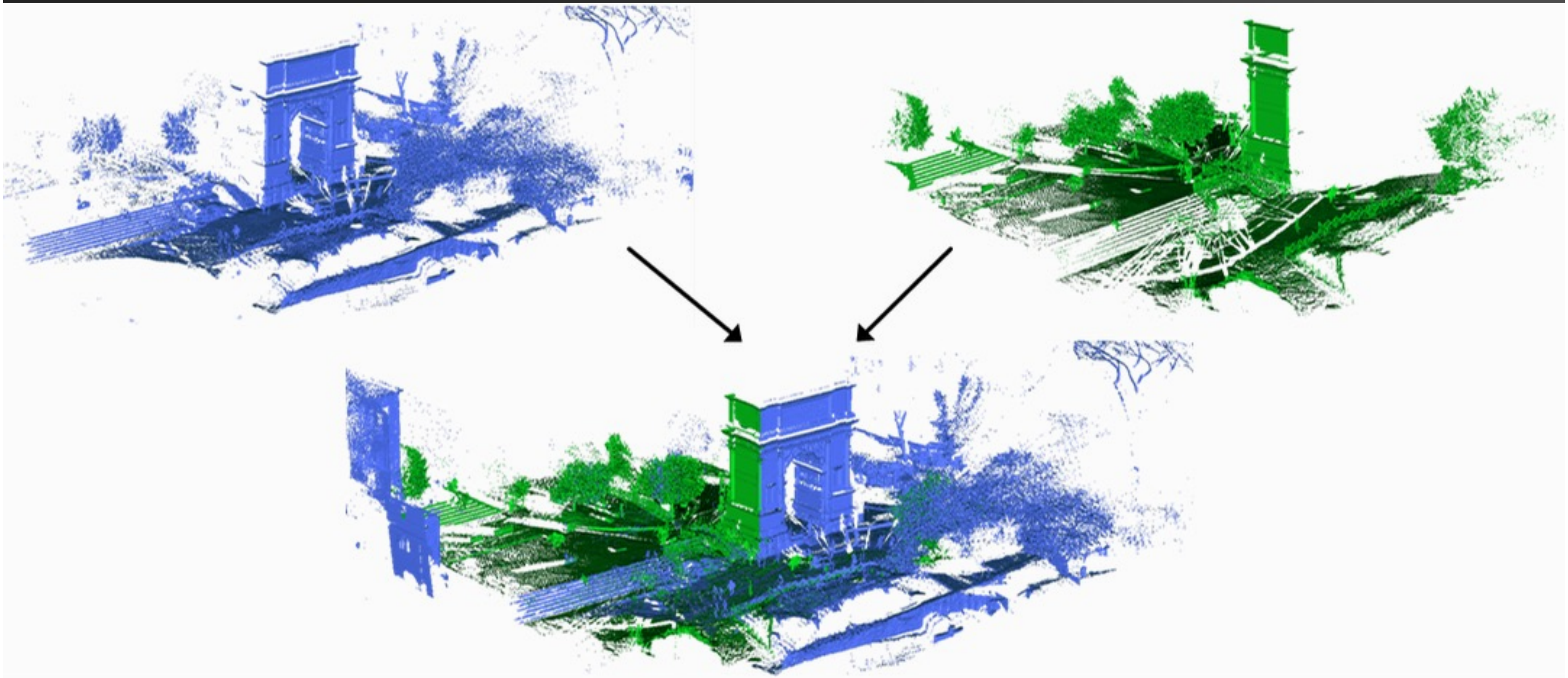
FastScan ($23K)

NextEngine (< $3K)

Roland DGA LPX-250 ($10K)

Point cloud acquisition

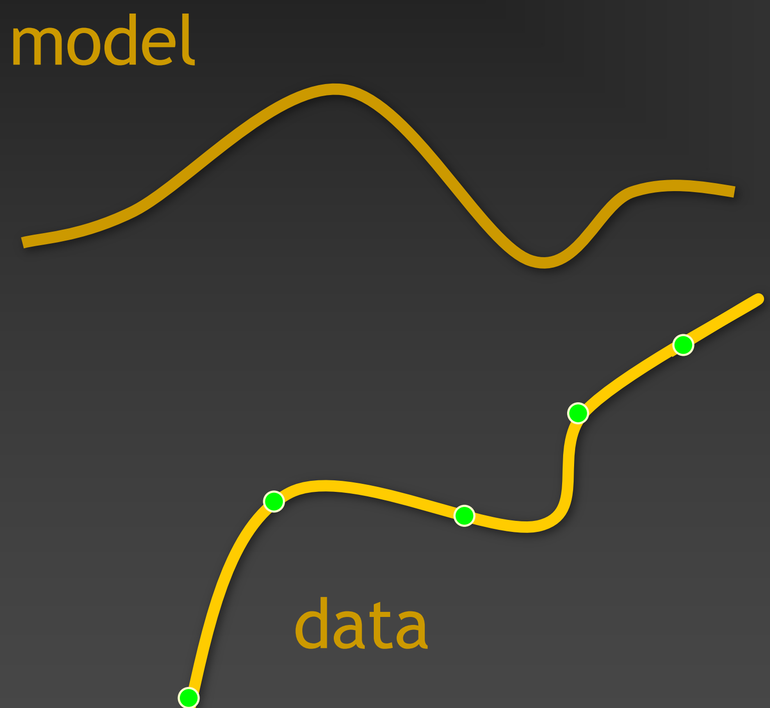
Point cloud registration



Iterative closest point (ICP) algorithm

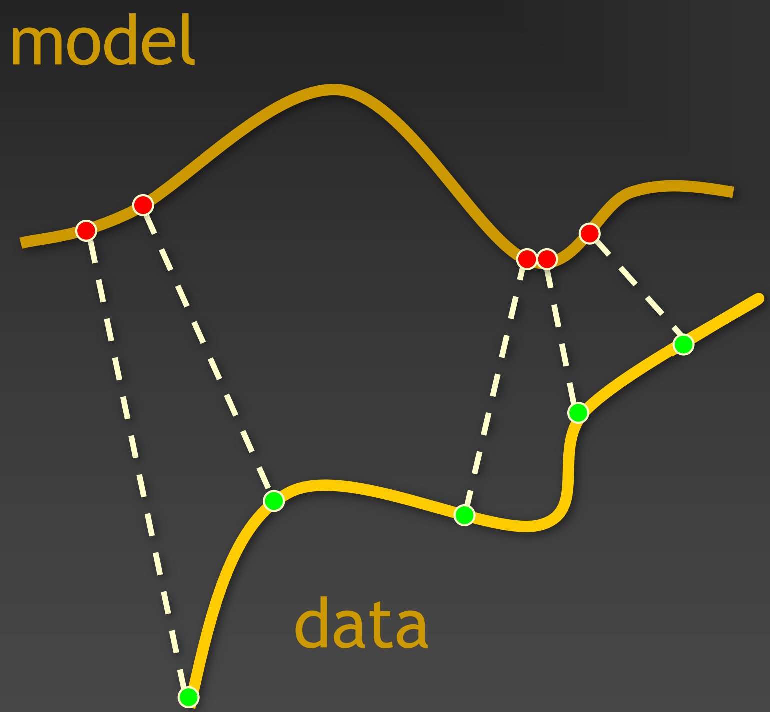
- A classic registration/correspondence schemes
- Input: **data** and **model shapes**
- Objective:
  - Rigid transform = rotation + translation
  - **Minimize mean squared error** from data points to closest points in model [Besl and Mckay 92]
- Correspondence obtained by Euclidean proximity

ICP algorithm



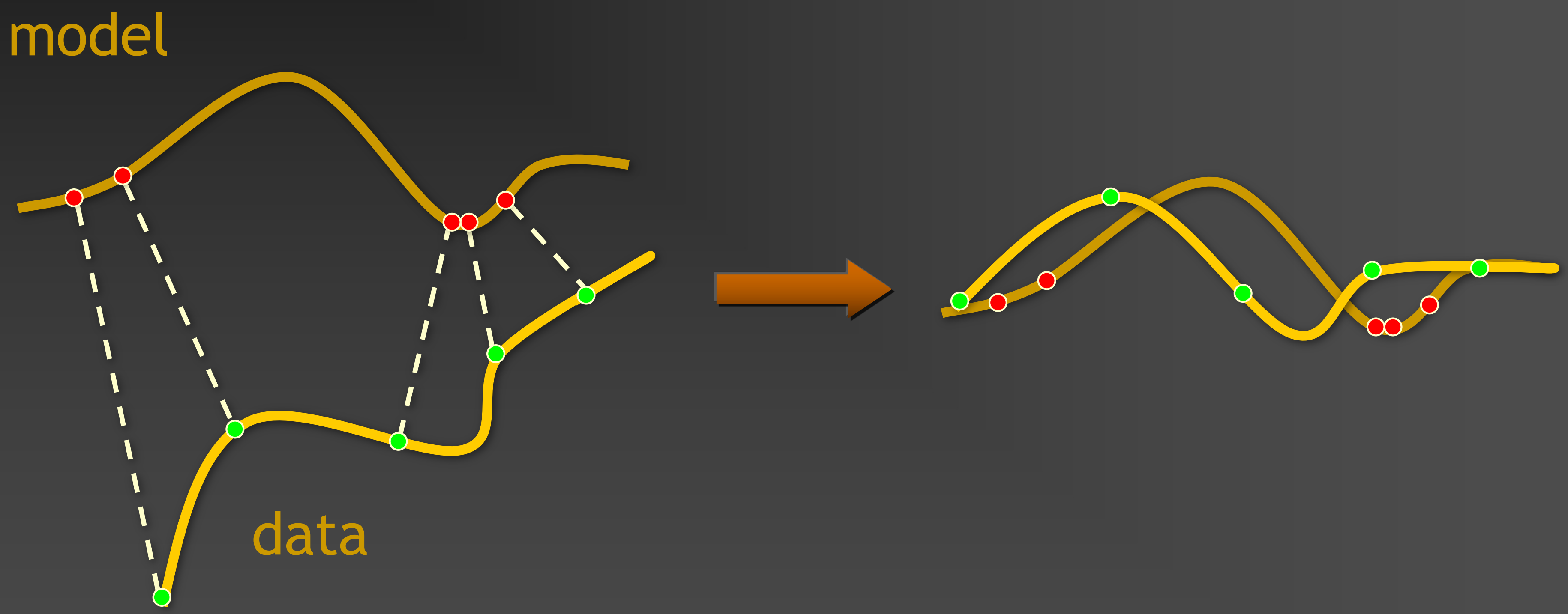
Model and data shapes (point samples)

ICP algorithm



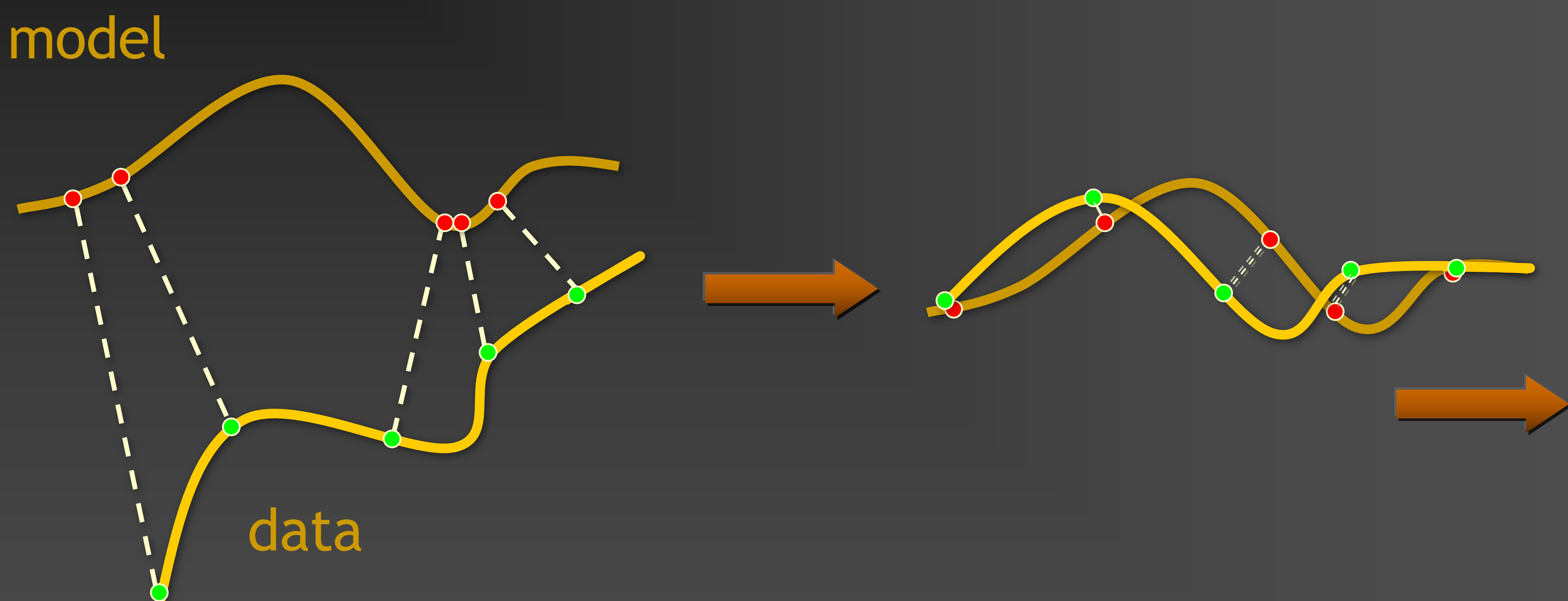
Find closest points from data to model

ICP algorithm



Find best rigid transform to align the corresponding points

ICP algorithm



Iterate …

A historical note on PBRs

- "As the visual complexity of computer-generated scenes continue to increase, the use of classical modeling primitives (polygons) as display primitives becomes less appealing."

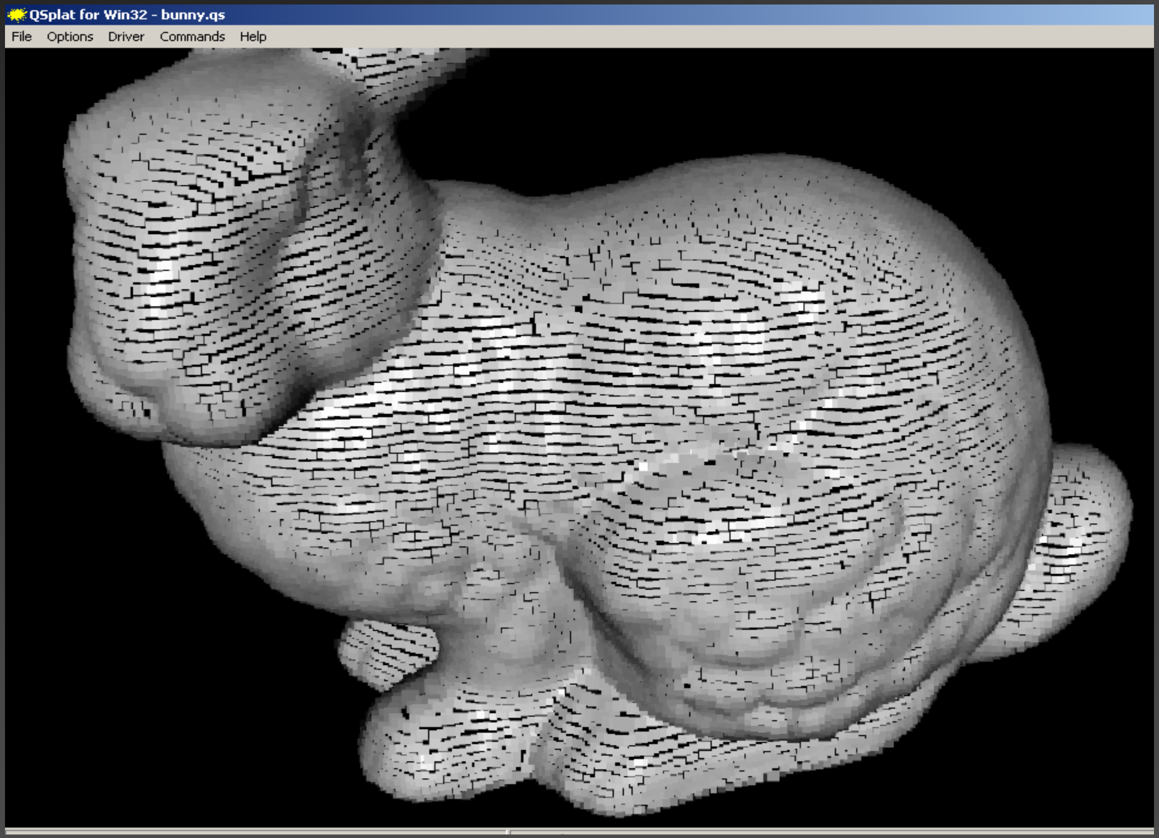
  Levoy and Whitted,
  "The Use of Points as a Display Primitive", 1985

- Use of points traces back to modeling of smoke, fire, and cloud around the late 70's [Csuri et al. 79, Blinn 82]

First renewed interest

- PBR was witnessing a revival around 2002 - 2012
- Points are directly available via **laser scanning**
  - Substantial advances in 3D digitizing and laser scanning and acquisition technology
  - High quality points (color and texture) easily obtained
  - Cheap scanners (< $3,000) available now

PBR rendering via splatting: QSplat



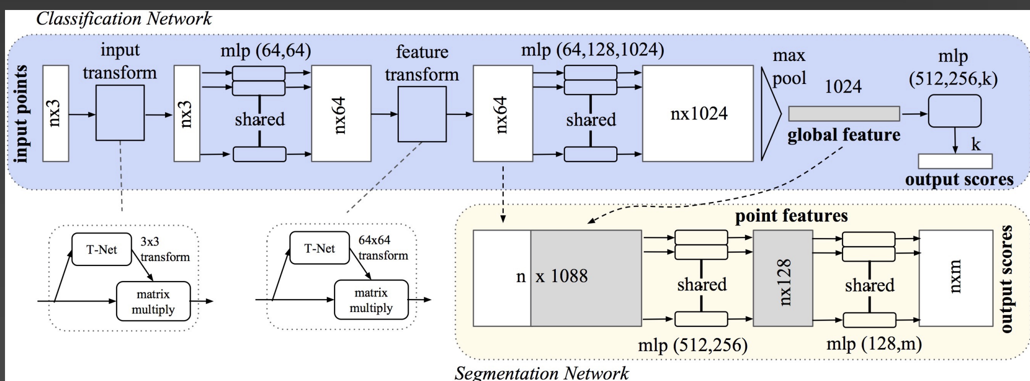
Splat = OpenGL points

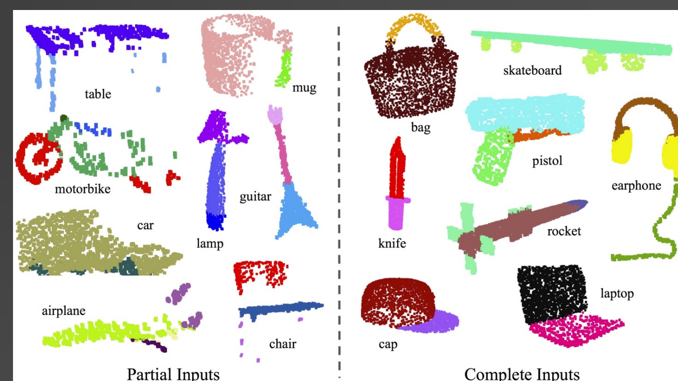
PBR rendering via splatting: QSplat

Splat = circles

Second revival: deep learning

- PointNet and PointNET++, since 2016/17
- Deep neural network to **encode and aggregate point features** for shape recognition, segmentation, etc.





Third revival: 3D Gaussian splatting (3DGS)

- 3DGS [Kerbl et al. SIGGRAPH 2023] superseding NeRF (2000)