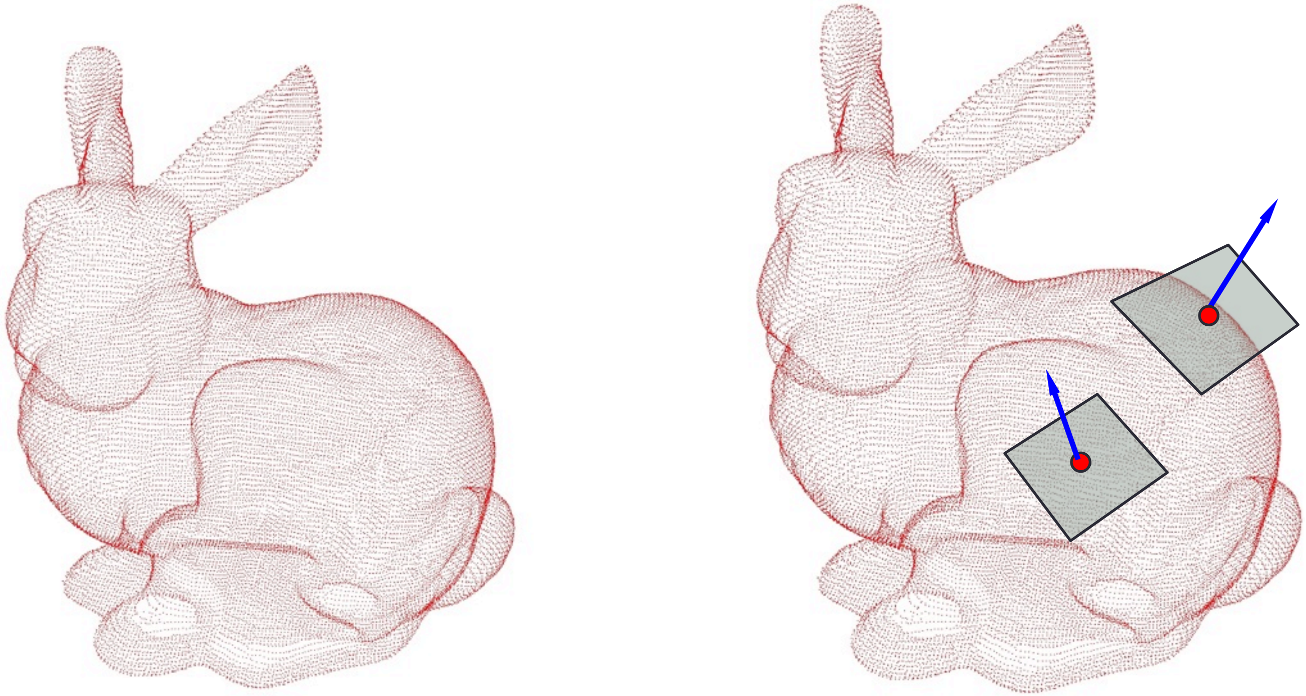
DIGITAL GEOMETRY PROCESSING

Algorithms for Representing, Analyzing and Comparing 3D shapes

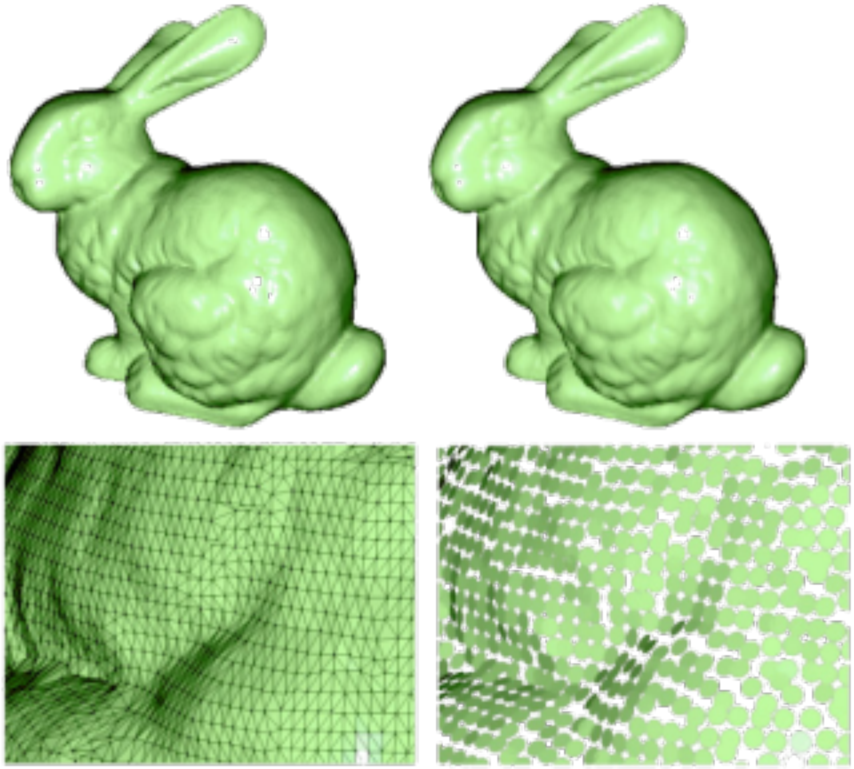
Point Clouds

- Simplest representation: **only points,** no connectivity.
- Collection of (x,y,z) coordinates, possibly with normals



Point Clouds

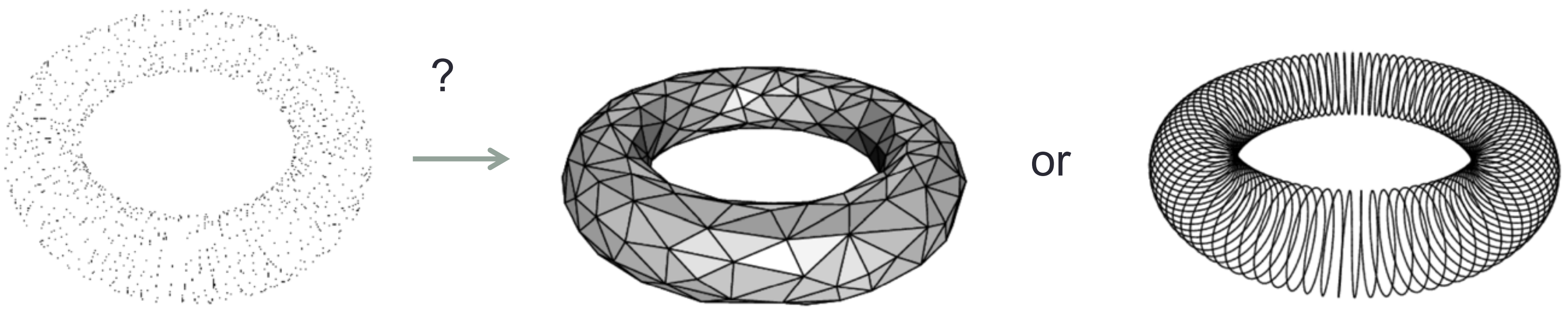
- Simplest representation: **only points,** no connectivity.
- Collection of (x,y,z) coordinates, possibly with normals.
- Points with orientation are called **surfels.**



Filip Van Bouwel

Point Clouds

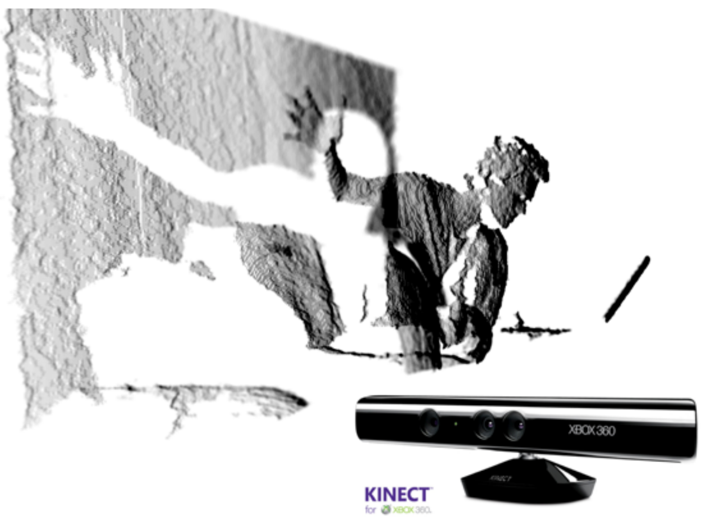
- Simplest representation: **only points,** no connectivity.
- Collection of (x,y,z) coordinates, possibly with normals.
- Points with orientation are called **surfels.**
- Severe limitations:
  - **no** Simplification or subdivision
  - **no** direct smooth rendering
  - **no** topological information

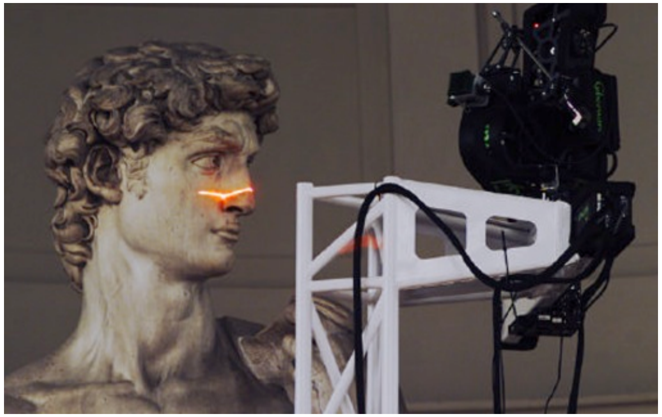


Why Point Clouds?

1) Typically, that's the only thing that's available

   Nearly all 3d scanning devices produce point clouds

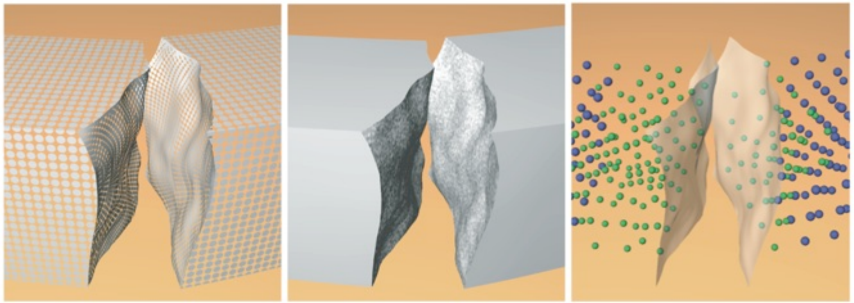




Why Point Clouds?

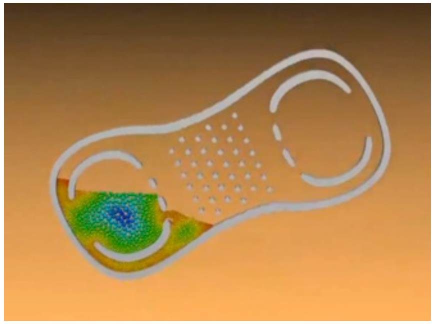
1) Typically, that's the only thing that's available
2) Locality: sometimes, easier to handle (esp. in hardware).

Fracturing Solids



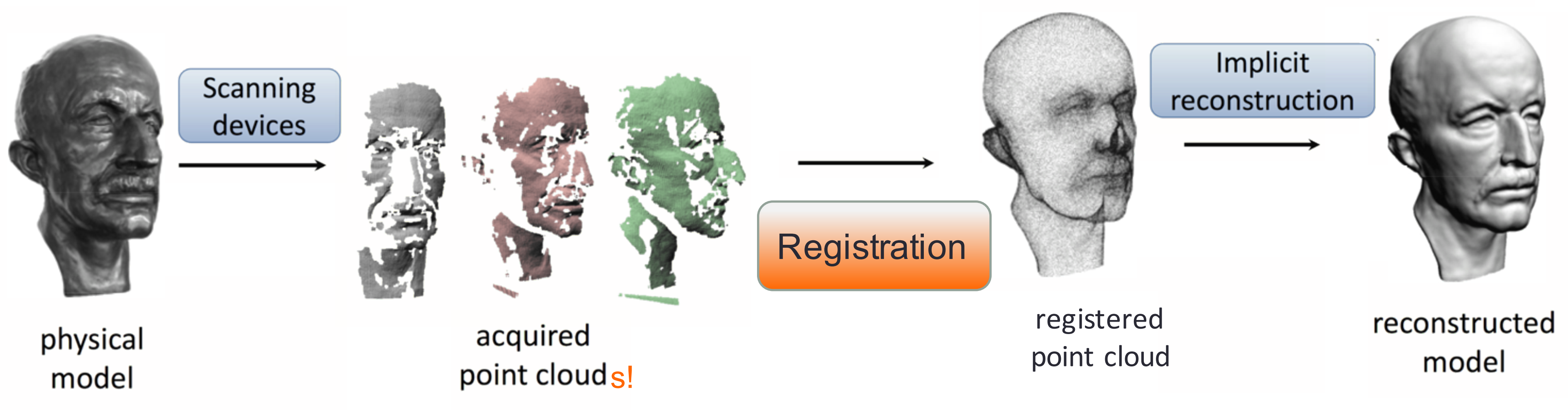
Meshless Animation of Fracturing Solids
Pauly et al., SIGGRAPH '05

Fluid Simulation



Adaptively sampled particle fluids,
Adams et al. SIGGRAPH '07

Typical Scanning and Reconstruction Pipeline



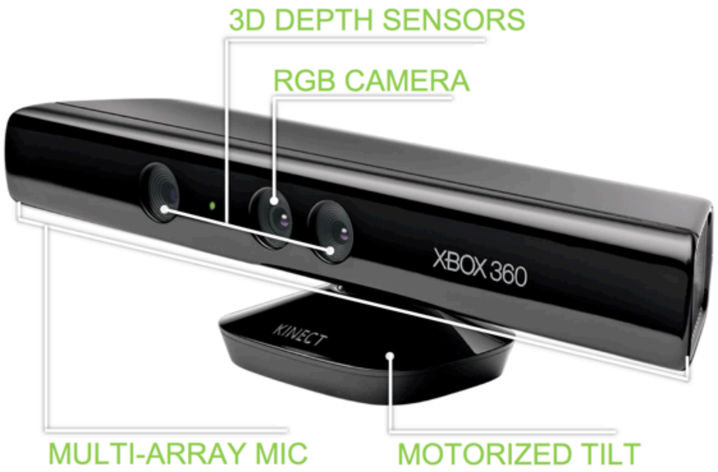
Single View Scanners

Major types of 3d scanners

- **Range (emission-based) scanners**
  - Time-of-flight laser scanner
  - Phase-based laser scanner
- **Triangulation**
  - Laser line sweep
  - Structured light
- **Stereo / computer vision**
  - Passive stereo
  - Active stereo / space time stereo

Microsoft Kinect 1 (2009)

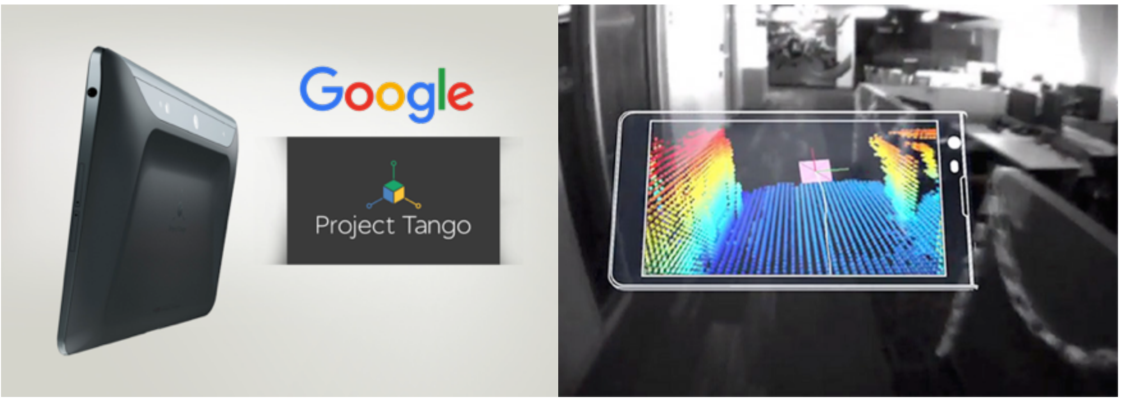
Low-cost (100$) 3d scanner – gadget for Xbox.



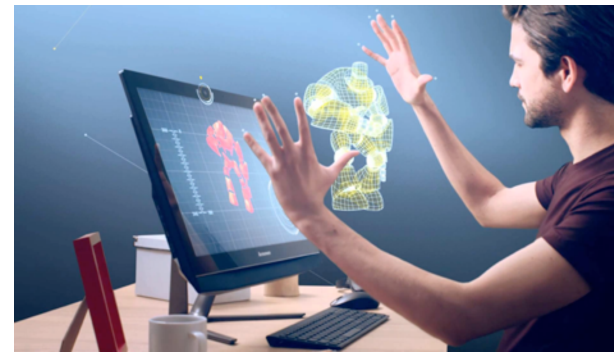
Allows to acquire Image (640 x 480) and 3d geometry (300k points) at 30 FPS.

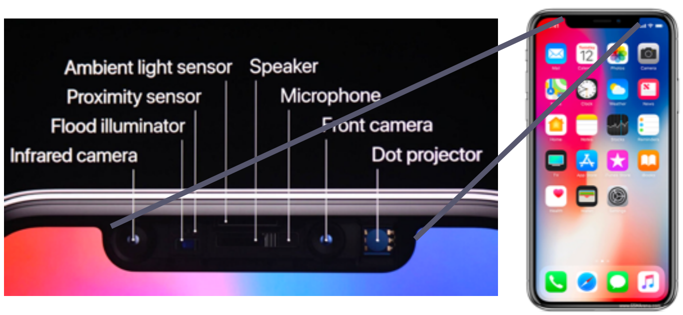
Uses infrared active illumination with an infrared sensor **and** depth-from blur.
accuracy of ~1mm (at 0.5m distance) to 4cm (at 2m distance).

Modern Mobile Devices (2017)

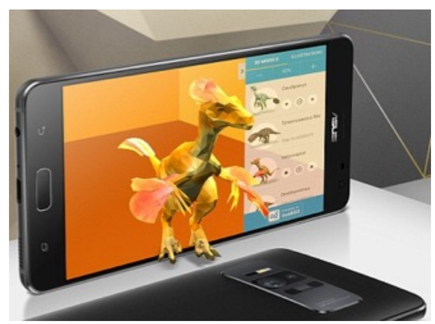


intel REALSENSE TECHNOLOGY

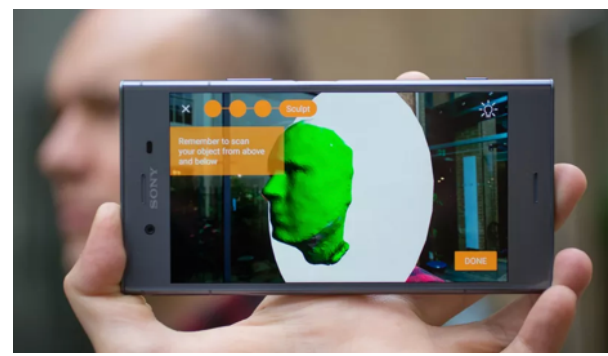




Apple iPhone X



Asus Zenfone AR



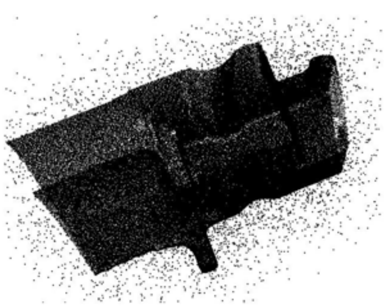
Sony Xperia XZ1

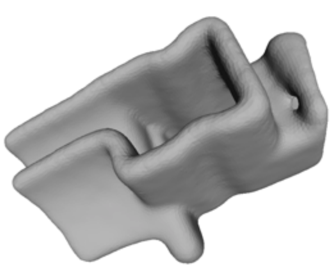
Typically use a combination of structured (infrared) light + stereo based depth.

3d Point Cloud Processing

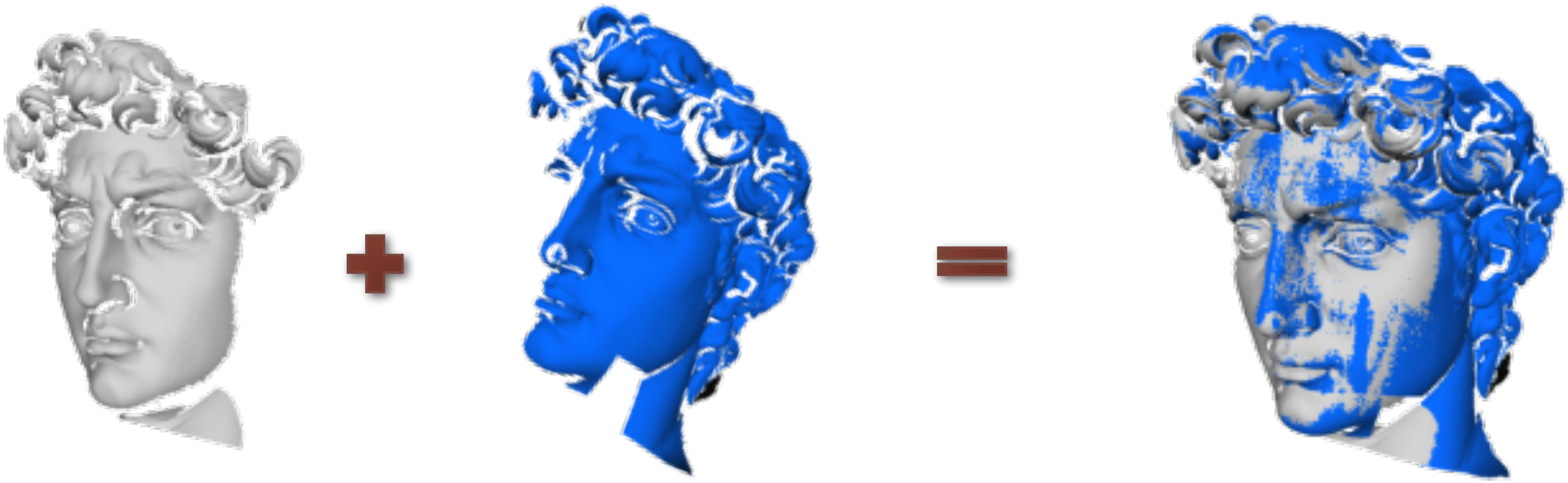
Typically point cloud sampling of a shape is insufficient for most applications. Main stages in processing:

1. Shape scanning (acquisition)
2. If have multiple scans, align them.
3. Smoothing – remove local noise.
4. Estimate surface normals.
5. Surface reconstruction
    - Implicit representation (today).
    - Triangle mesh (today).





Fundamental Registration Problem



Given (at least) two shapes with partially overlapping geometry, find an alignment between them.

Why Registration?

Fundamental problem in geometry analysis

Appears in many shape analysis applications

ICP: one of the best-known algorithms in computer graphics and computational geometry. Widely used in industry.

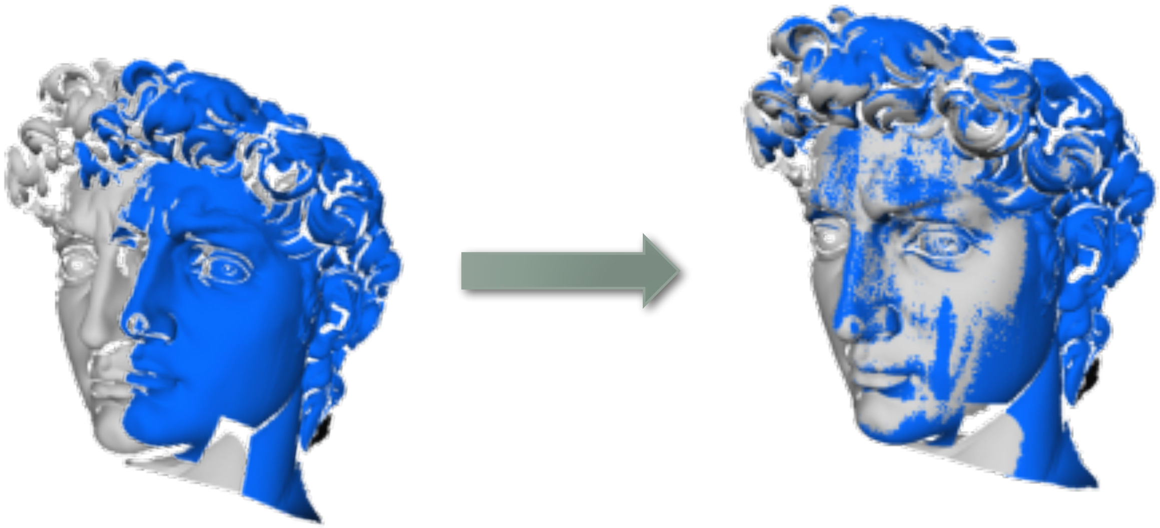
For you: very nice programming exercise.

Quick introduction to an active research area.

…

Local Alignment

- Simplest instance of the registration problem



Given two shapes that are **approximately aligned** (e.g. by a human) we want to find the optimal tranformation.

Other Applications

- Manufacturing:

  One shape is a **model** and the other is a **scan** of a product. Finding defects.

- Medicine:

  Finding correspondences between 3D MRI scans of the same person or different people.

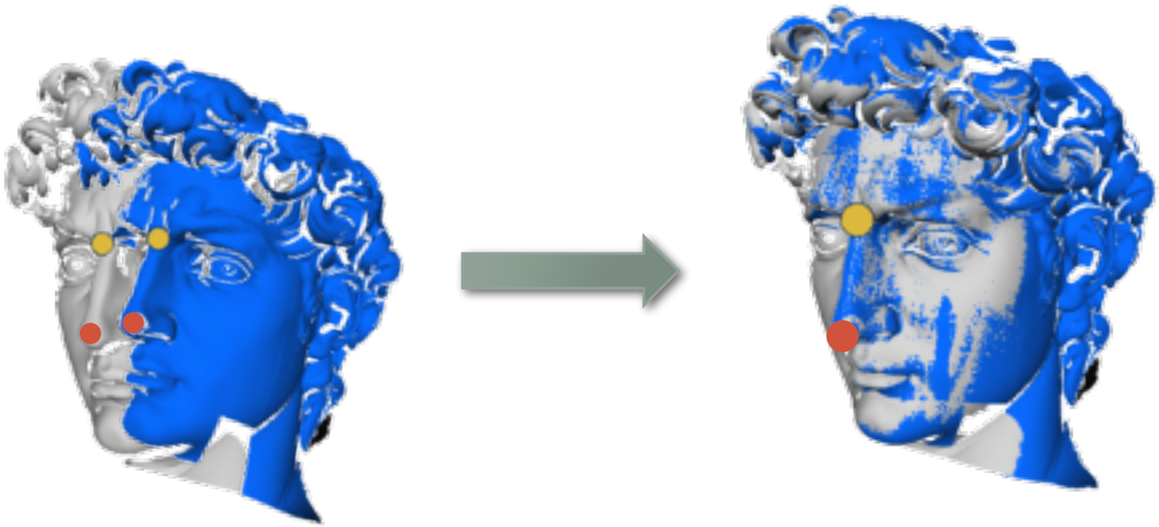
- Animation Reconstruction & 3D Video.

- Statistical Shape Analysis:

  Building models for a collection of shapes.

Local Alignment

- What does it mean for an alignment to be **good**?



Intuition: want corresponding points to be close after transformation.

Problems
1. We don't know what points correspond.
2. We don't know the optimal alignment.

Iterative Closest Point (ICP)

- Approach: iterate between finding correspondences and finding the transformation:

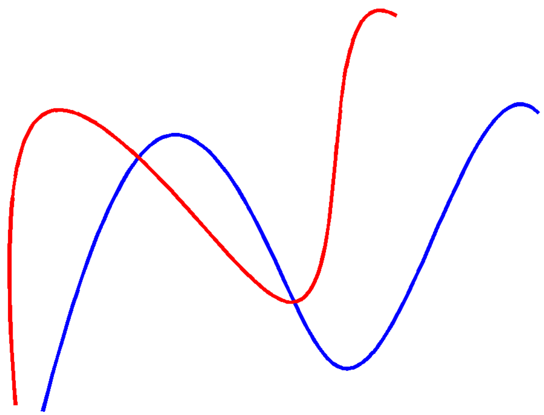
Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing:

$$\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point (ICP)

- Approach: iterate between finding correspondences and finding the transformation:

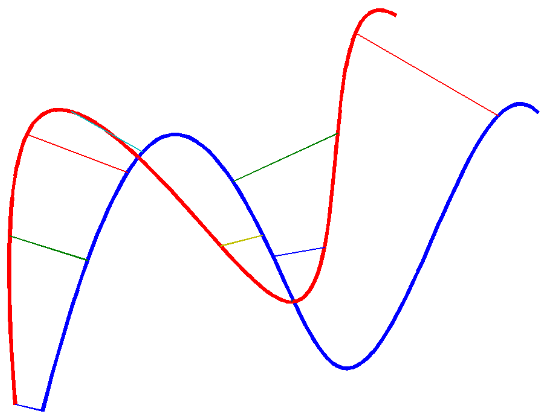


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

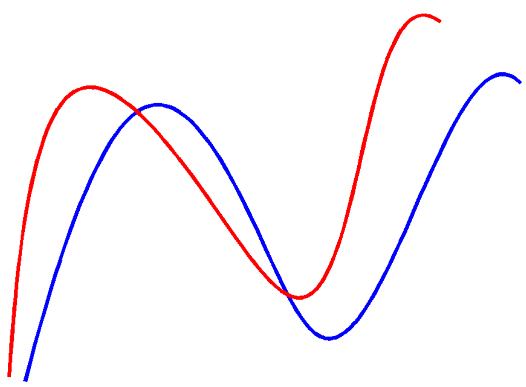


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

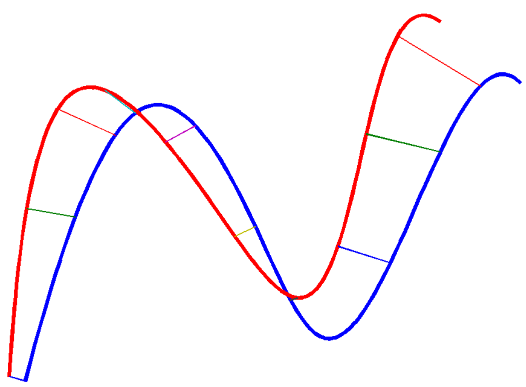


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

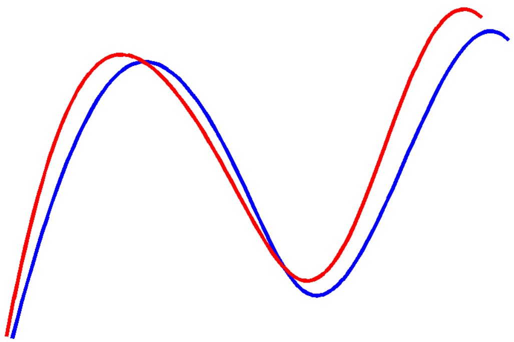


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

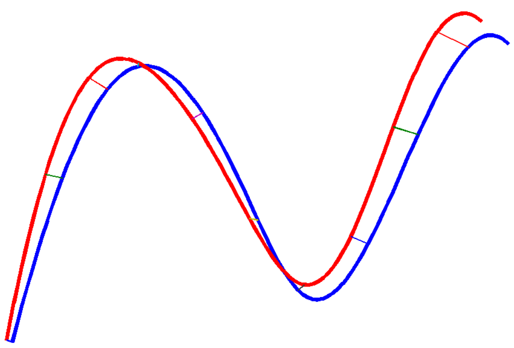


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

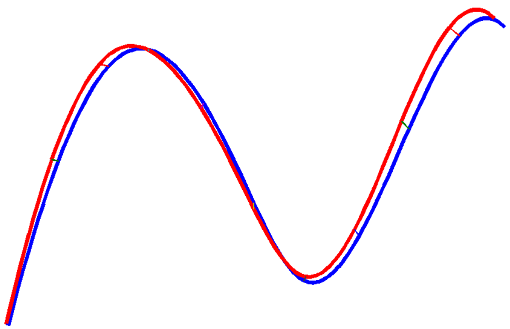


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

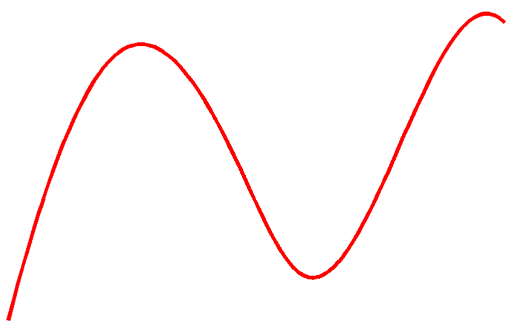


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:



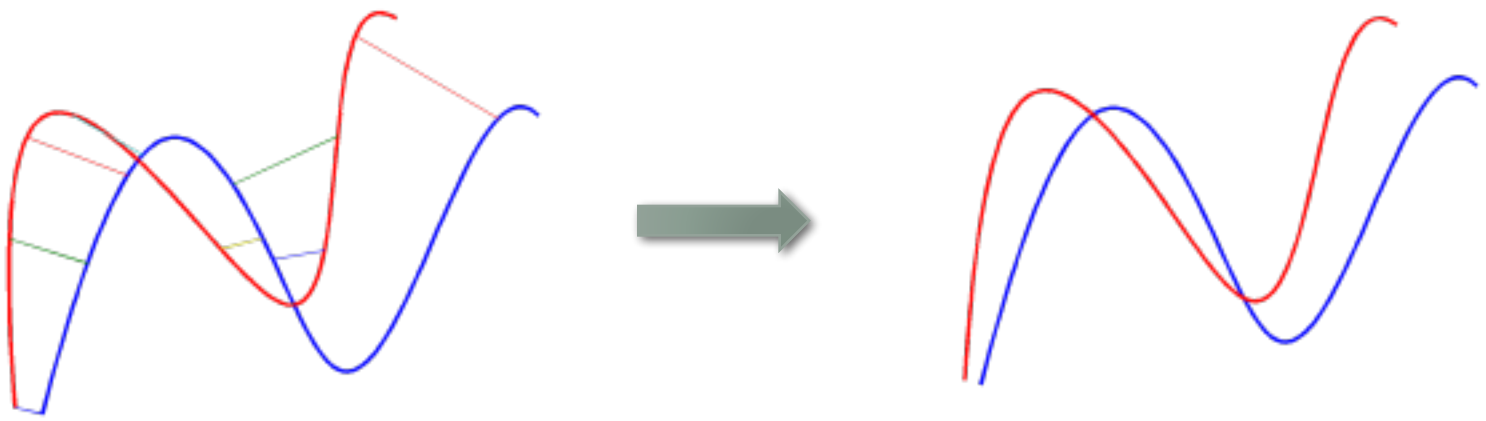
Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Requires two main computations:

1. Computing nearest neighbors.
2. Computing the optimal transformation



ICP: Nearest Neighbor Computation

Closest points

$$y_i = \arg\min_{y \in Y} \|y - x_i\|$$

- How to find closest points **efficiently**?
- Straightforward complexity: $\mathcal{O}(MN)$

  M number of points on X, N number of points on Y.

- Y divides the space into **Voronoi cells**

$$V(y \in Y) = \{z \in \mathbb{R}^3 : \|y - z\| < \|y' - z\| \; \forall \; y' \in Y \neq y\}$$

- Given a query point y, **determine to which cell it belongs**.

ICP: Nearest Neighbor Computation

Closest points

$$y_i = \arg\min_{y \in Y} \|y - x_i\|$$

- How to find closest points **efficiently**?
- Straightforward complexity: $\mathcal{O}(MN)$

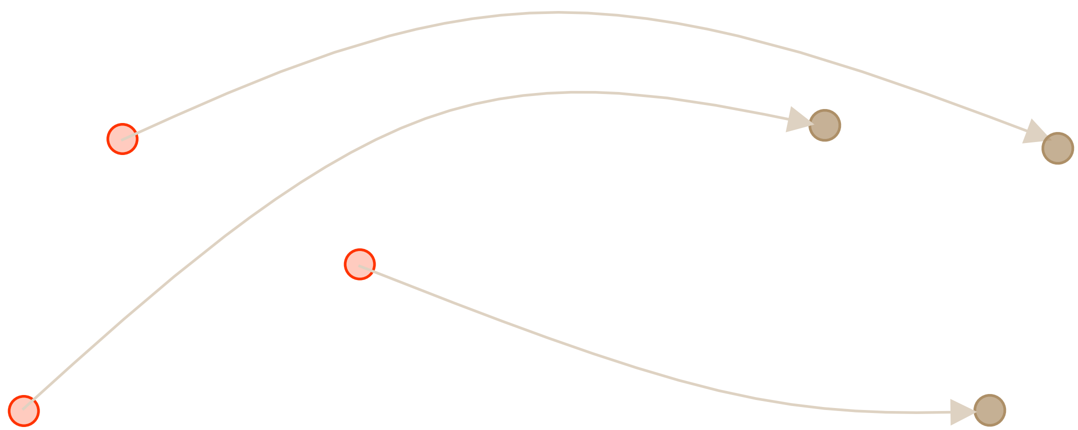
  M number of points on X, N number of points on Y.

ICP: Optimal Transformation

Problem Formulation:

1. Given two sets points: $\{x_i\}, \{y_i\}, i = 1..n$ in $\mathbb{R}^3$. Find the rigid transform: $\mathbf{R}, t$ that minimizes:

$$\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$$



ICP: Optimal Transformation

Problem Formulation:

1. Given two sets points: $\{x_i\}, \{y_i\}, i = 1..n$ in $\mathbb{R}^3$. Find the rigid transform: $\mathbf{R}, t$ that minimizes:

$$\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$$

ICP: Optimal Transformation

Problem Formulation:

1. Given two sets points: $\{x_i\}, \{y_i\}, i = 1..n$ in $\mathbb{R}^3$. Find the rigid transform: $\mathbf{R}, t$ that minimizes: $$\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$$
2. Closed form solution with rotation matrices:
   1. Construct: $C = \sum_{i=1}^{N} (y_i - \mu^Y)(x_i - \mu^X)^T$ where $\mu^X = \frac{1}{N}\sum_i x_i,$ $\mu^Y = \frac{1}{N}\sum_i y_i$
   2. Compute the SVD of C: $C = U\Sigma V^T$
      1. If $\det(UV^T) = 1, R_{\text{opt}} = UV^T$
      2. Else $R_{\text{opt}} = U\tilde{\Sigma}V^T, \tilde{\Sigma} = \text{diag}(1, 1, \ldots, -1)$
   3. Set $t_{\text{opt}} = \mu^Y - R_{\text{opt}}\mu^X$

Note that C is a 3x3 matrix. SVD is very fast.

Arun et al., Least-Squares Fitting of Two 3-D Point Sets

Iterative Closest Point.

Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Convergence:

- at each iteration $\sum_{i=1}^{N} d^2(x_i, Y)$ decreases.
- Converges to local minimum
- Good initial guess: global minimum.

[Besl&McKay92]

Variations of ICP

1. Selecting source points (from one or both scans): sampling
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs
5. Assigning an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation