5.2 **Surface Registration**

Hao Li

http://cs621.hao-li.com

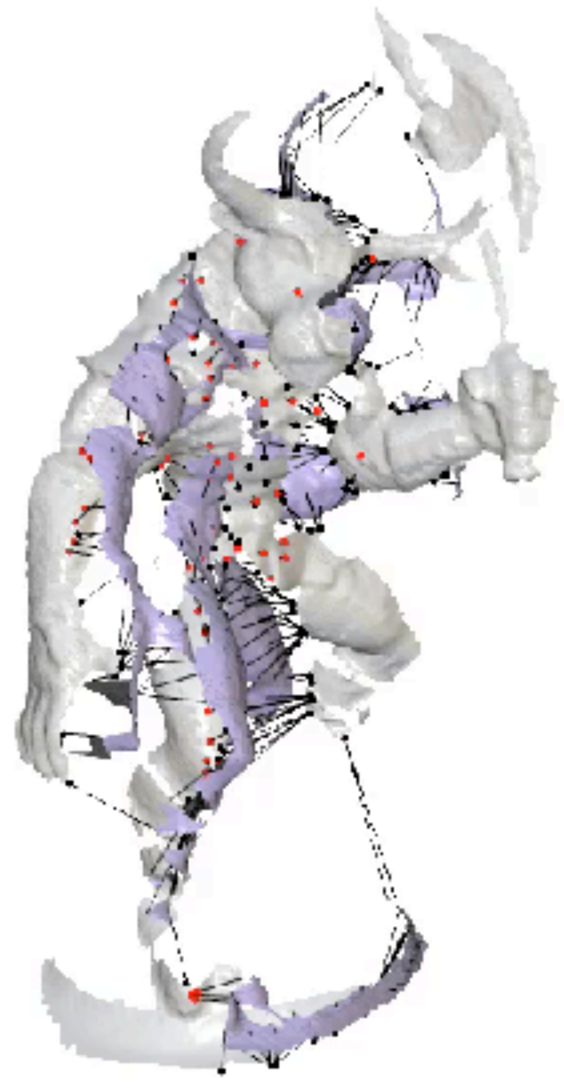
Acknowledgement

Images and Slides are courtesy of

- Prof. Szymon Rusinkiewicz, Princeton University
- ICCV Course 2005: http://www.cis.upenn.edu/~bjbrown/iccv05_course/

Surface Registration

Align two partially-overlapping meshes given initial guess for relative transform

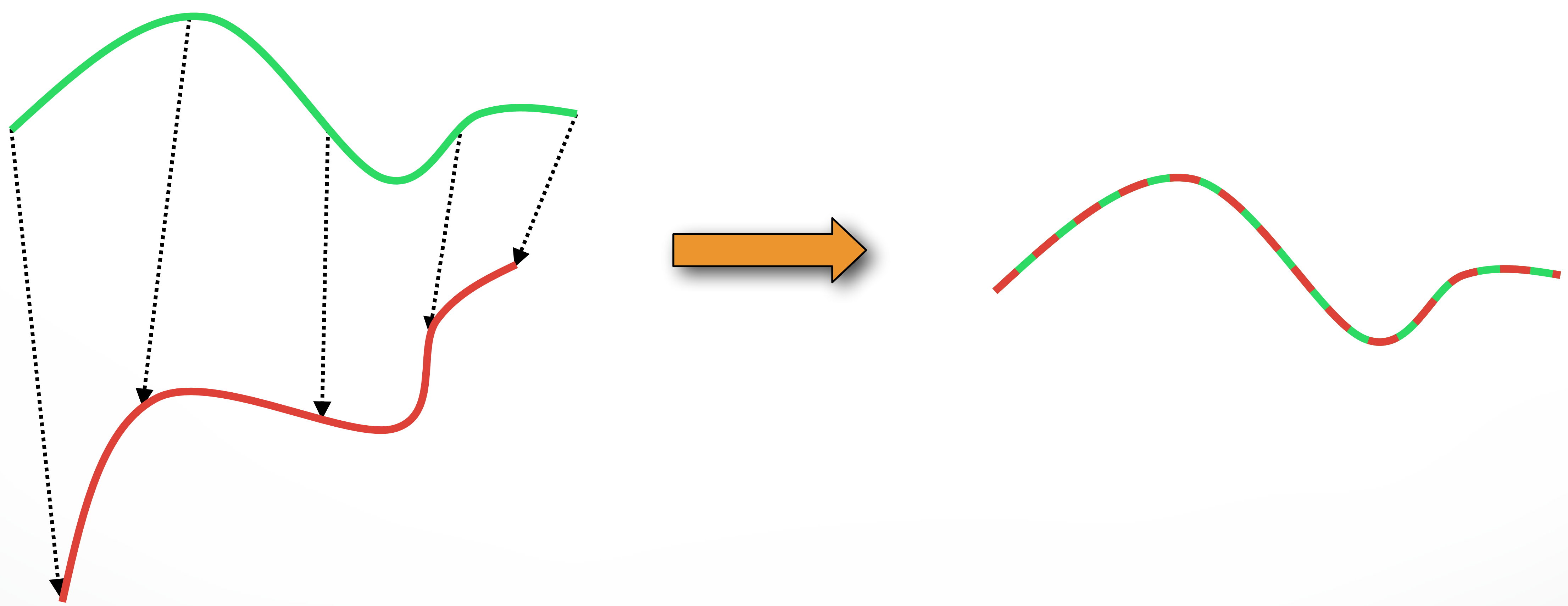


Outline

- ICP: Iterative Closest Points
- Classification of ICP variants
  - Faster alignment
  - Better robustness
- ICP as function minimization

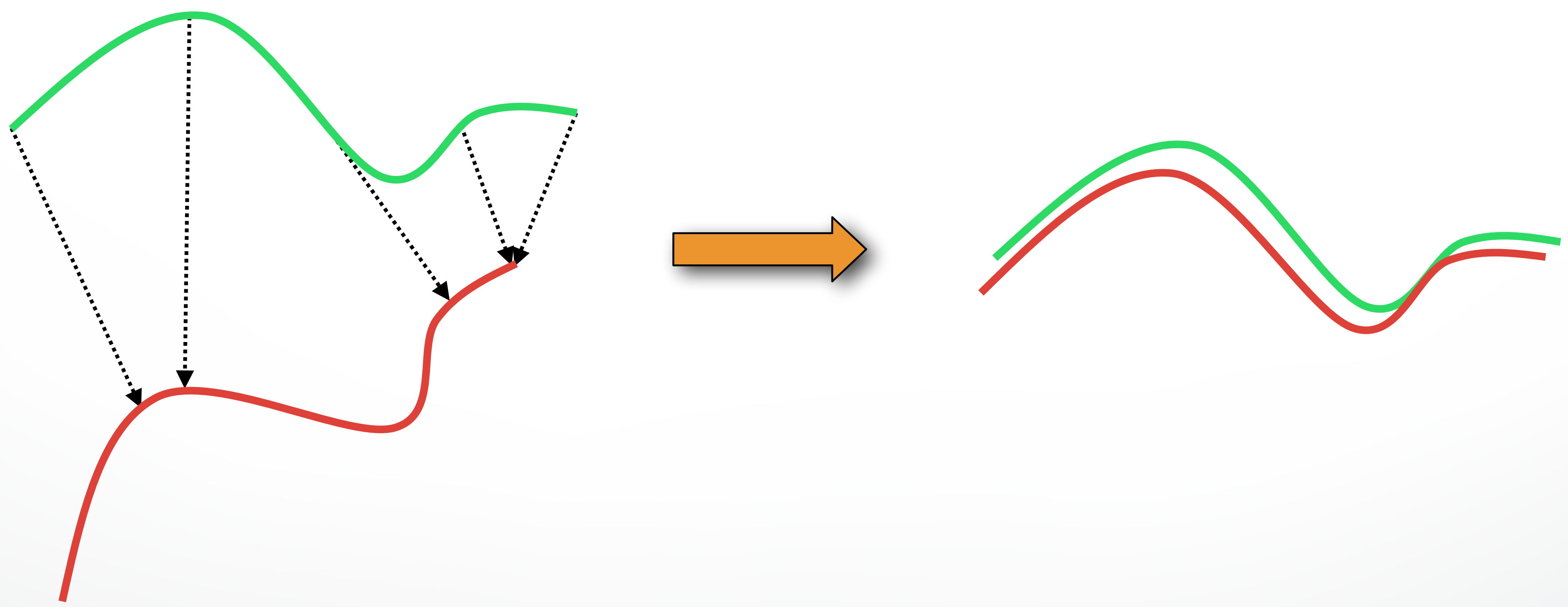
Aligning 3D Data

If correct correpondences are known, can find correct relative rotation/translation



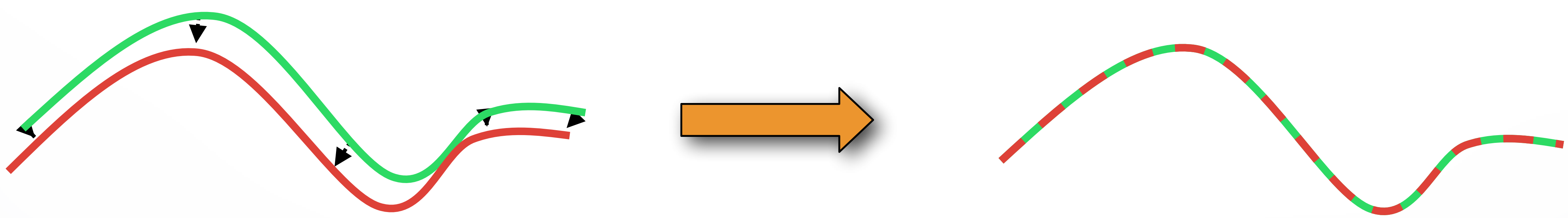
Aligning 3D Data

- How to find correspondences: User input? Feature detection? Signatures?
- Alternatives: assume **closest** points correspond



Aligning 3D Data

- … and iterate to find alignment
  - Iterative Closest Points (ICP) [Besl & Mckay]
- Converges if starting position "close enough"



Basic ICP

- **Select** e.g., 1000 random points
- **Match** each to closest point on other scan, using data structure such as *k*-d tree
- **Reject** pairs with distance > *k* times median
- Construct **error function**:

$$E = \sum \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2$$

- **Minimize** (closed form solution in [Horn 87])

Shape Matching: Translation

- Define bary-centered point sets

$$
\begin{aligned}
\bar{\mathbf{p}} &:= \frac{1}{m}\sum_{i=1}^{m}\mathbf{p}_i & \qquad \bar{\mathbf{q}} &:= \frac{1}{m}\sum_{i=1}^{m}\mathbf{q}_i \\
\hat{\mathbf{p}}_i &:= \mathbf{p}_i - \bar{\mathbf{p}} & \qquad \hat{\mathbf{q}}_i &:= \mathbf{q}_i - \bar{\mathbf{q}}
\end{aligned}
$$

- Optimal translation vector $\mathbf{t}$ maps barycenters onto each other

$$\mathbf{t} = \bar{\mathbf{p}} - \mathbf{R}\bar{\mathbf{q}}$$

Shape Matching: Rotation

- Approximate nonlinear rotation by general matrix

$$\min_{\mathbf{R}} \sum_i \|\hat{\mathbf{p}}_i - \mathbf{R}\hat{\mathbf{q}}_i\|^2 \quad \rightarrow \quad \min_{\mathbf{A}} \sum_i \|\hat{\mathbf{p}}_i - \mathbf{A}\hat{\mathbf{q}}_i\|^2$$

- The least squares linear transformation is

$$\mathbf{A} = \left(\sum_{i=1}^{m} \hat{\mathbf{p}}_i \hat{\mathbf{q}}_i^T\right) \cdot \left(\sum_{i=1}^{m} \hat{\mathbf{q}}_i \hat{\mathbf{q}}_i^T\right)^{-1} \in \mathbb{R}^{3\times 3}$$

- SVD & Polar decomposition extracts rotation from $\mathbf{A}$

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \rightarrow \quad \mathbf{R} = \mathbf{U}\mathbf{V}^T$$

ICP Variants

Variants on the following stages of ICP have been proposed

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

ICP Variants

Can analyze various aspects of performance:

- Speed
- Stability
- Tolerance of noise and/or outliers
- Maximum initial misalignment

Comparisons of many variants in

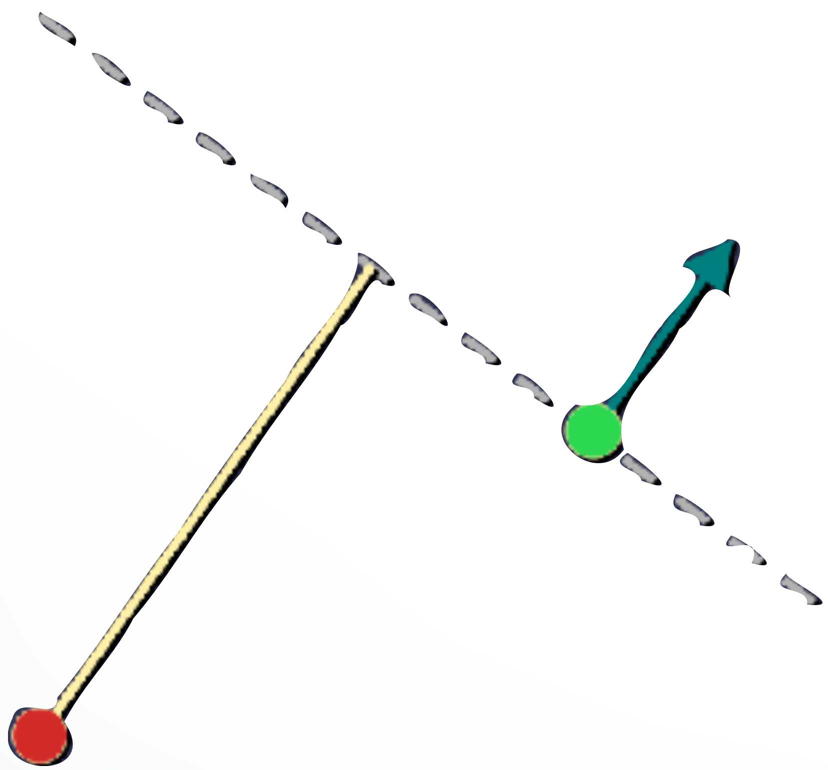
- [Rusinkiewicz & Levoy, 3DIM 2001]

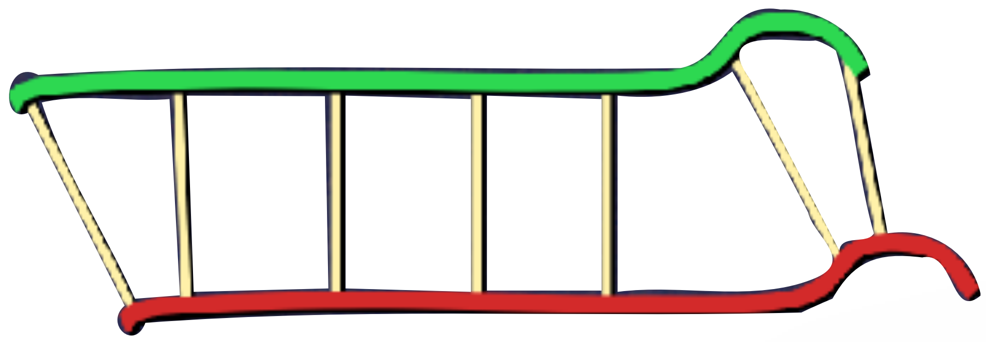
ICP Variants

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

Point-to-Plane Error Metric

Using point-to-plane distance instead of point-to-point lets flat regions slide along each other [Chen & Medioni 91]





Point-to-Plane Error Metric

- Error function:

$$E = \sum \left((\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)^\top \mathbf{n_i} \right)^2$$

where $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ is a translation vector

- Linearize (i.e. assume that $\sin\theta \approx \theta$, $\cos\theta \approx 1$):

$$E \approx \sum \left((\mathbf{p}_i - \mathbf{q}_i)^\top \mathbf{n}_i \right) \; + \mathbf{r}^\top (\mathbf{p}_i \times \mathbf{n}_i) + \mathbf{t}^\top \mathbf{n}_i)^2 \qquad \mathbf{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

- Result: overconstrained linear system

Point-to-Plane Error Metric

- Overconstrained linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A} = \begin{bmatrix} \leftarrow & \mathbf{p}_1 \times \mathbf{n}_1 & \rightarrow & \leftarrow & \mathbf{n}_1 & \rightarrow \\ \leftarrow & \mathbf{p}_2 \times \mathbf{n}_1 & \rightarrow & \leftarrow & \mathbf{n}_2 & \rightarrow \\ & \vdots & & & \vdots & \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ t_x \\ t_y \\ t_z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -(\mathbf{p}_1 - \mathbf{q}_1)^\top \mathbf{n}_1 \\ -(\mathbf{p}_2 - \mathbf{q}_2)^\top \mathbf{n}_2 \\ \vdots \end{bmatrix}$$

- Solve using least squares

$$\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}$$

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$$

Improving ICP Stabilitiy

- Closest **compatible** point
- Stable sampling

ICP Variants

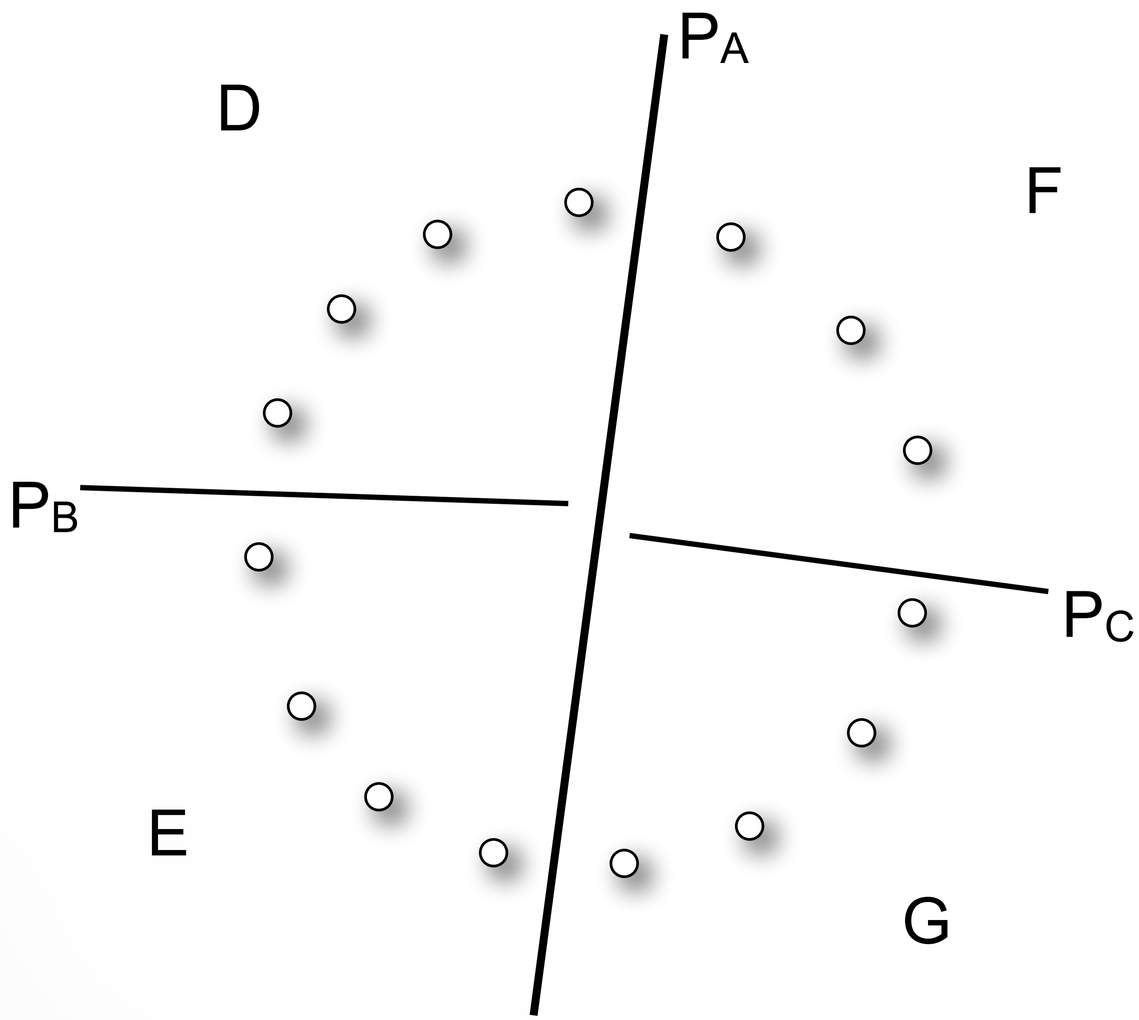
1. Selecting source points (from one or both meshes)
2. **Matching to points in the other mesh**
3. Weighting the correspondences
4. Rejecting certain (outliers) point pairs
5. Assining an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation

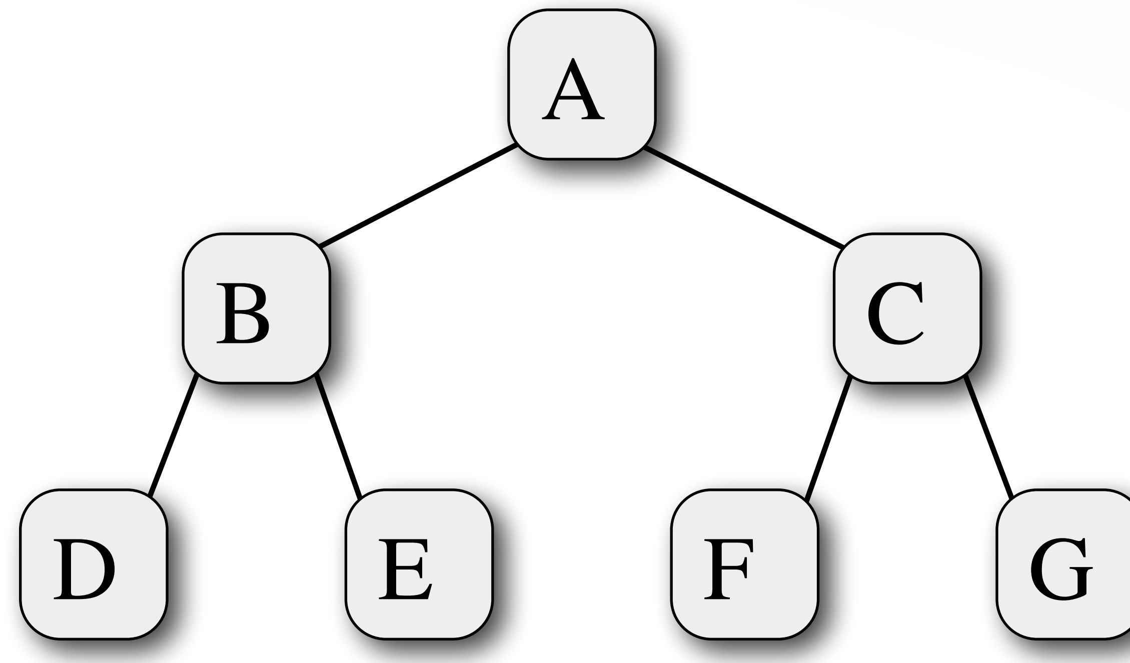
Closest Point Search

- Find closest point of a query point
  - Brute force: $O(n)$ complexity

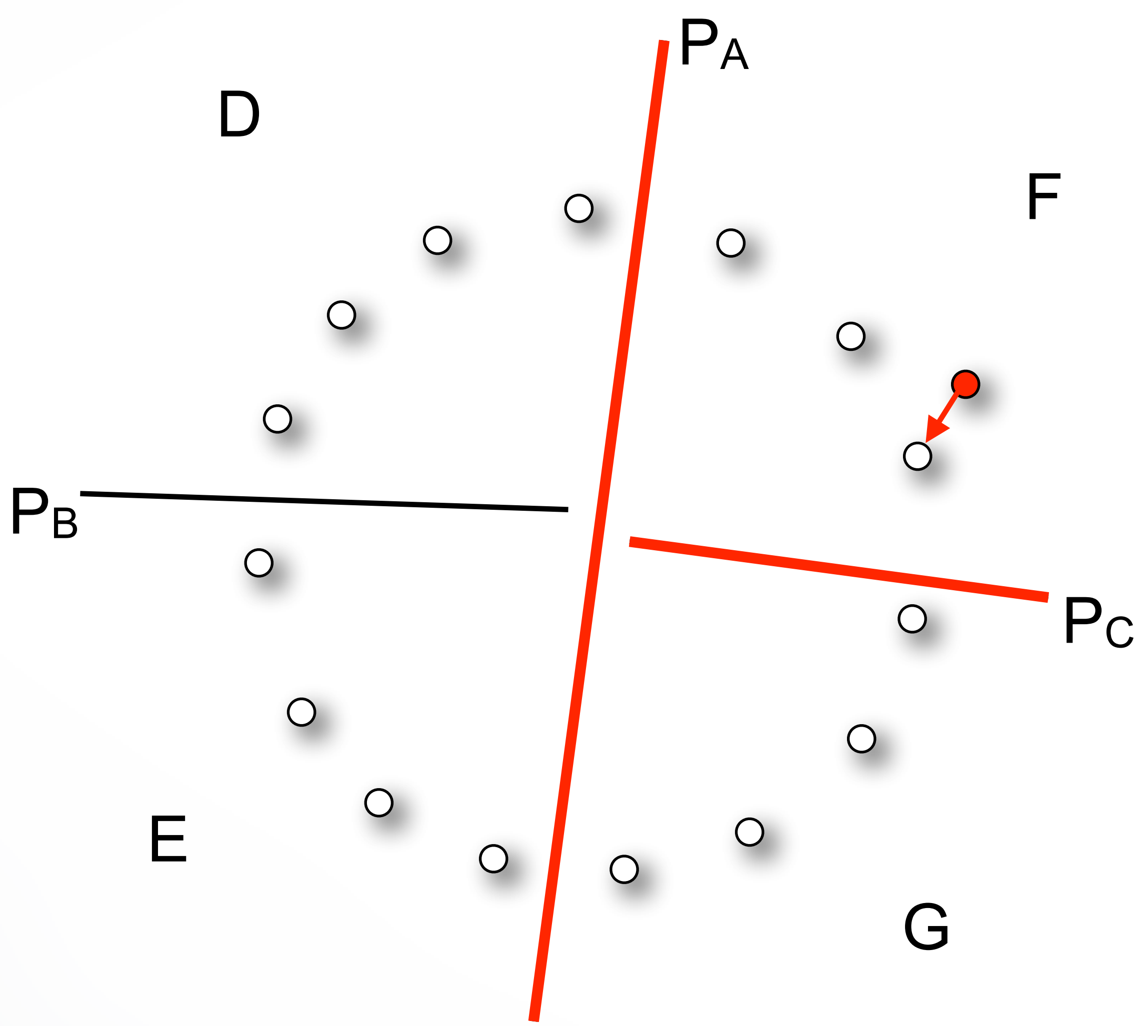
- Use Hierarchical BSP tree
  - Binary space partitioning tree (general kD-tree)
  - Recursively partition 3D space by planes
  - Tree should be balanced, put plane at median
  - $\log(n)$ tree levels, complexity $O(n\log n)$

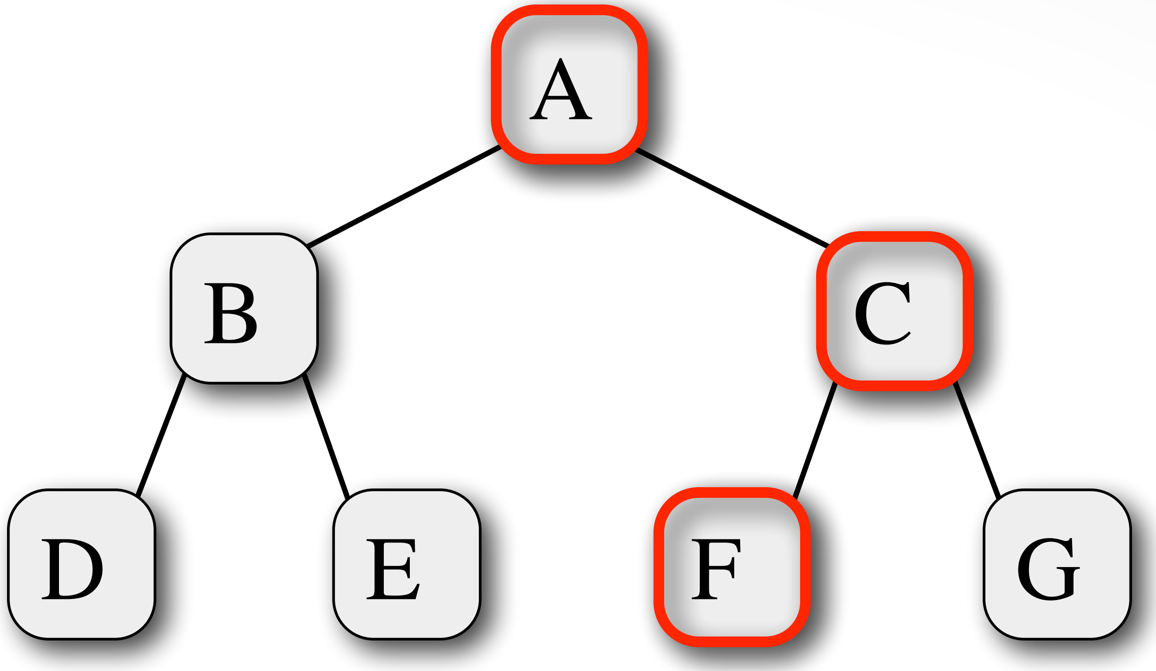
BSP Closest Point Search



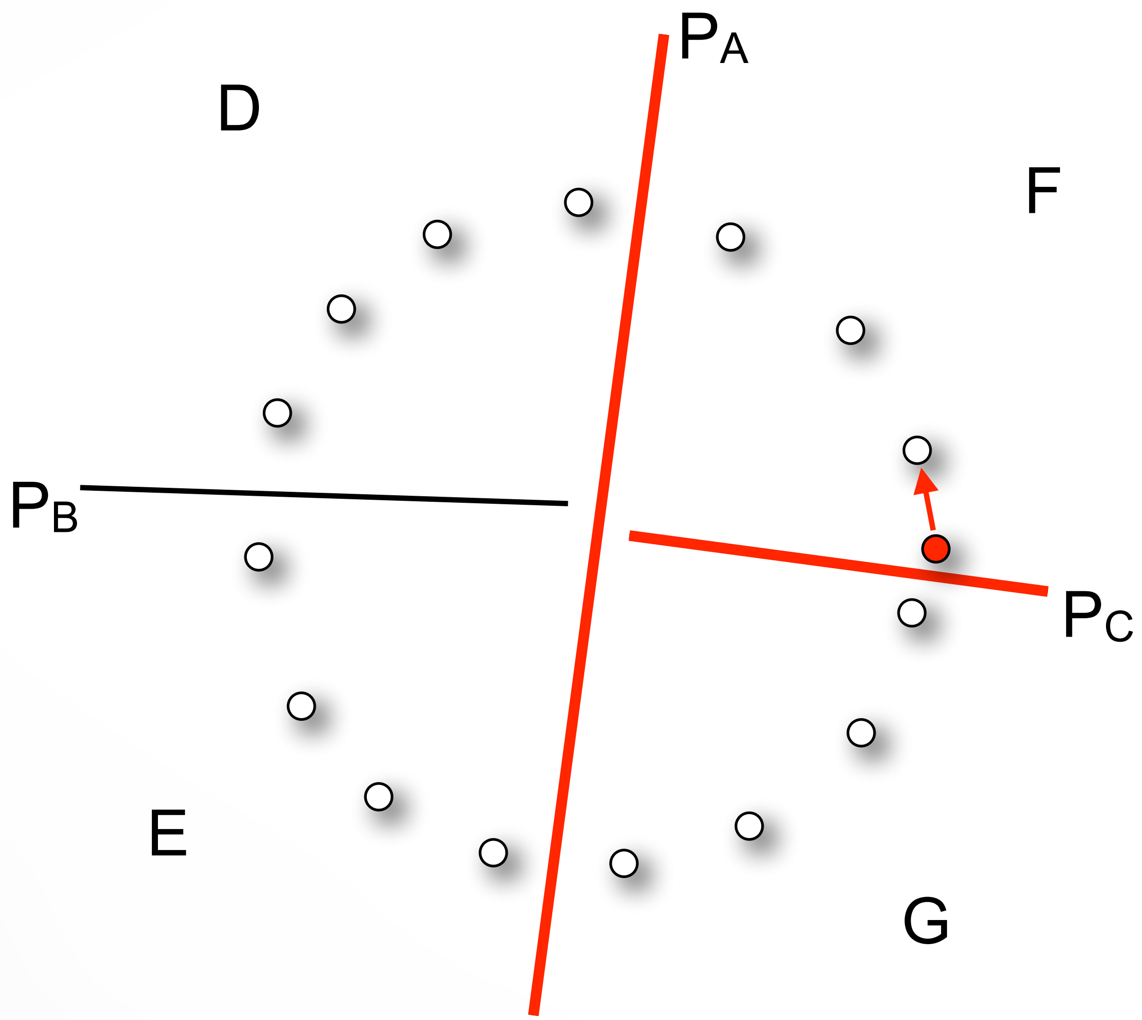


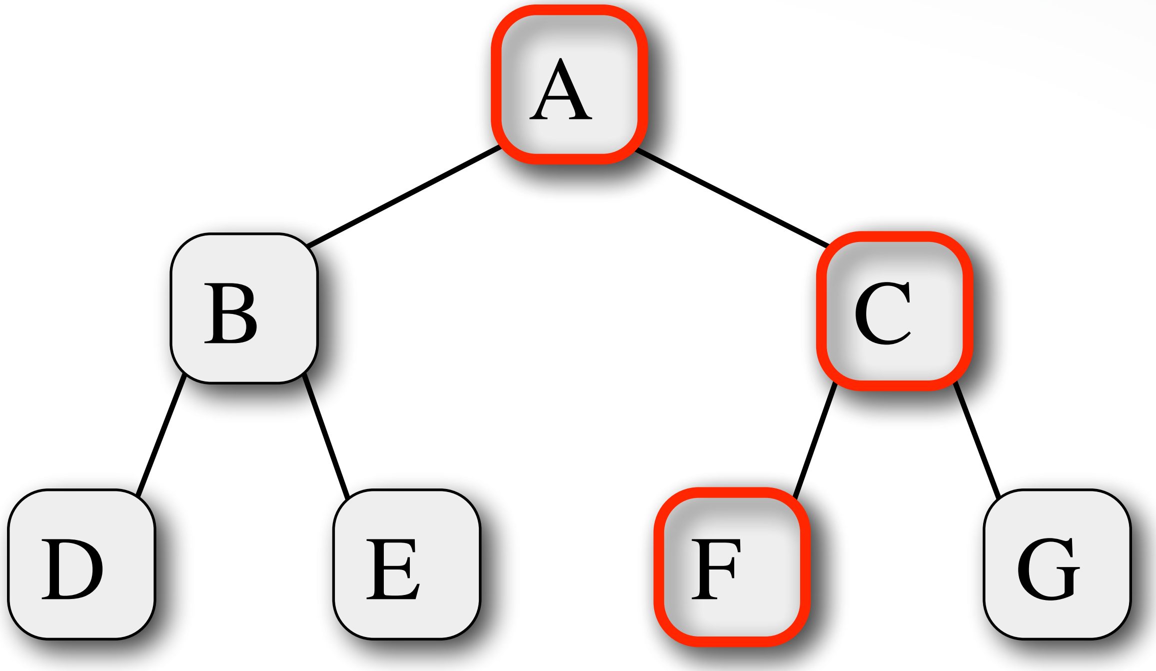
BSP Closest Point Search



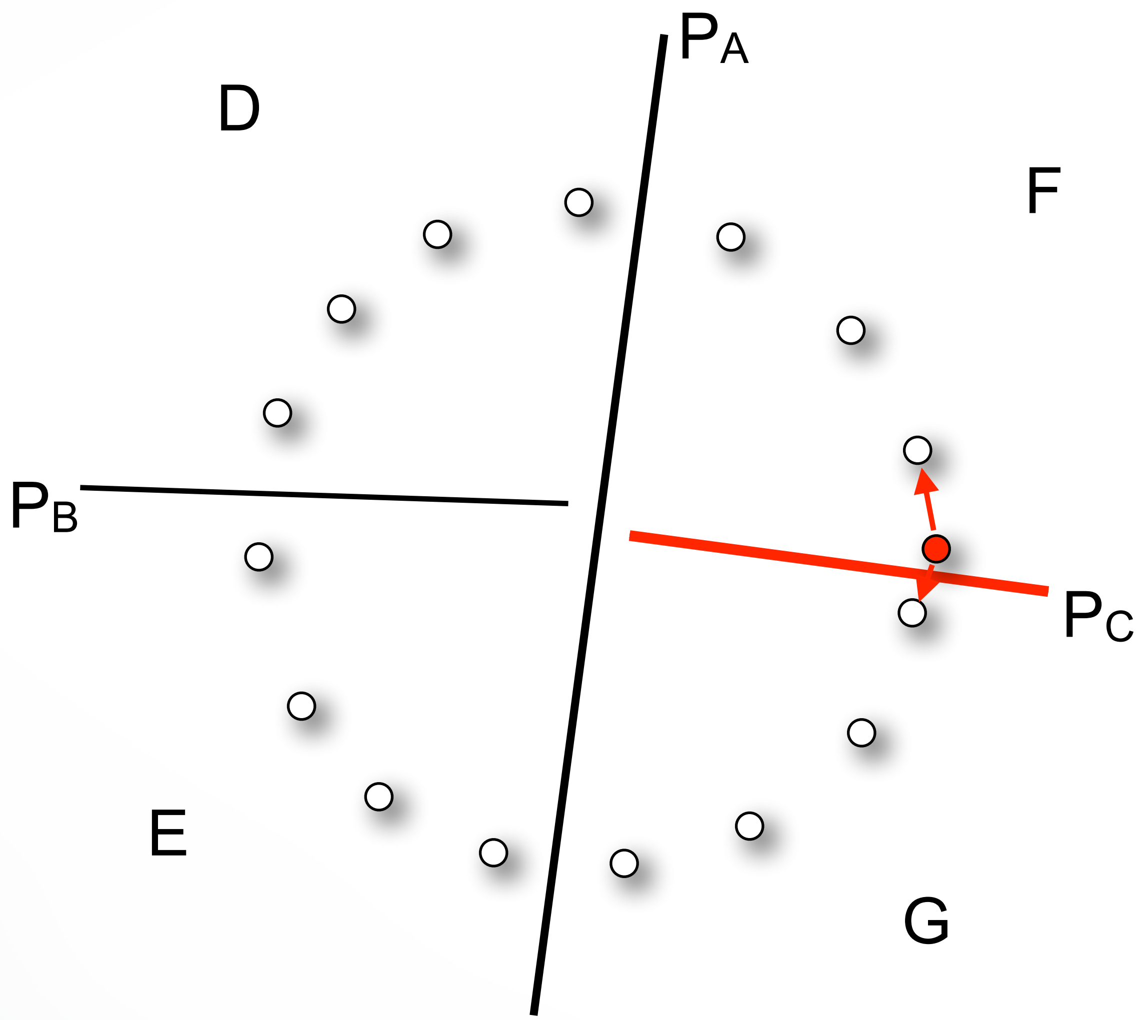


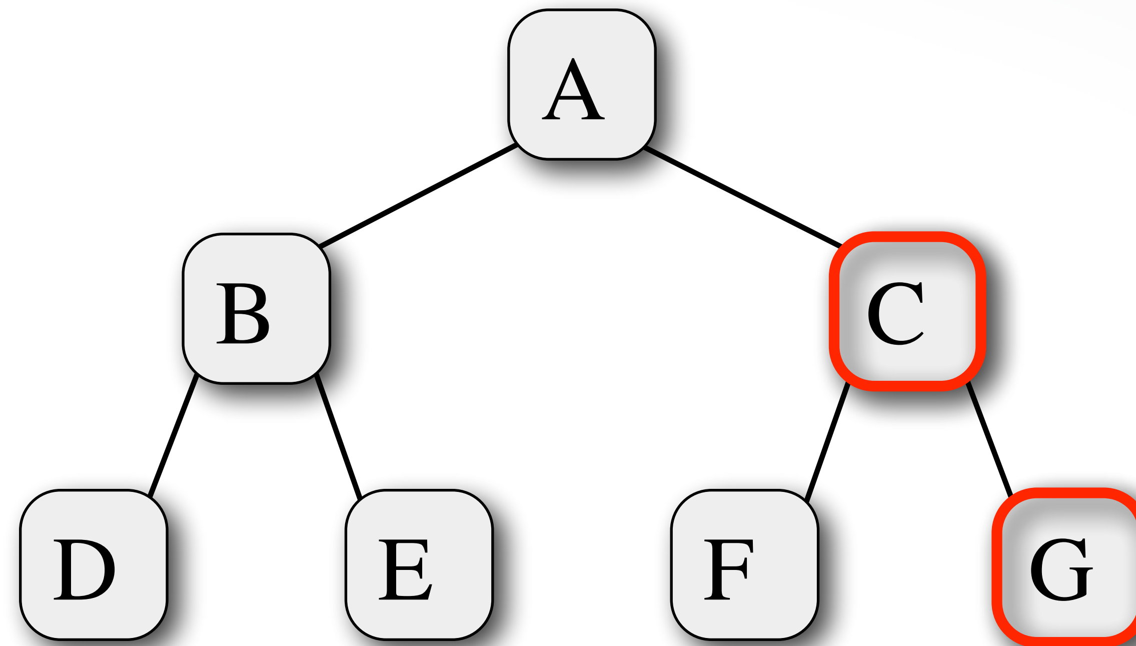
BSP Closest Point Search



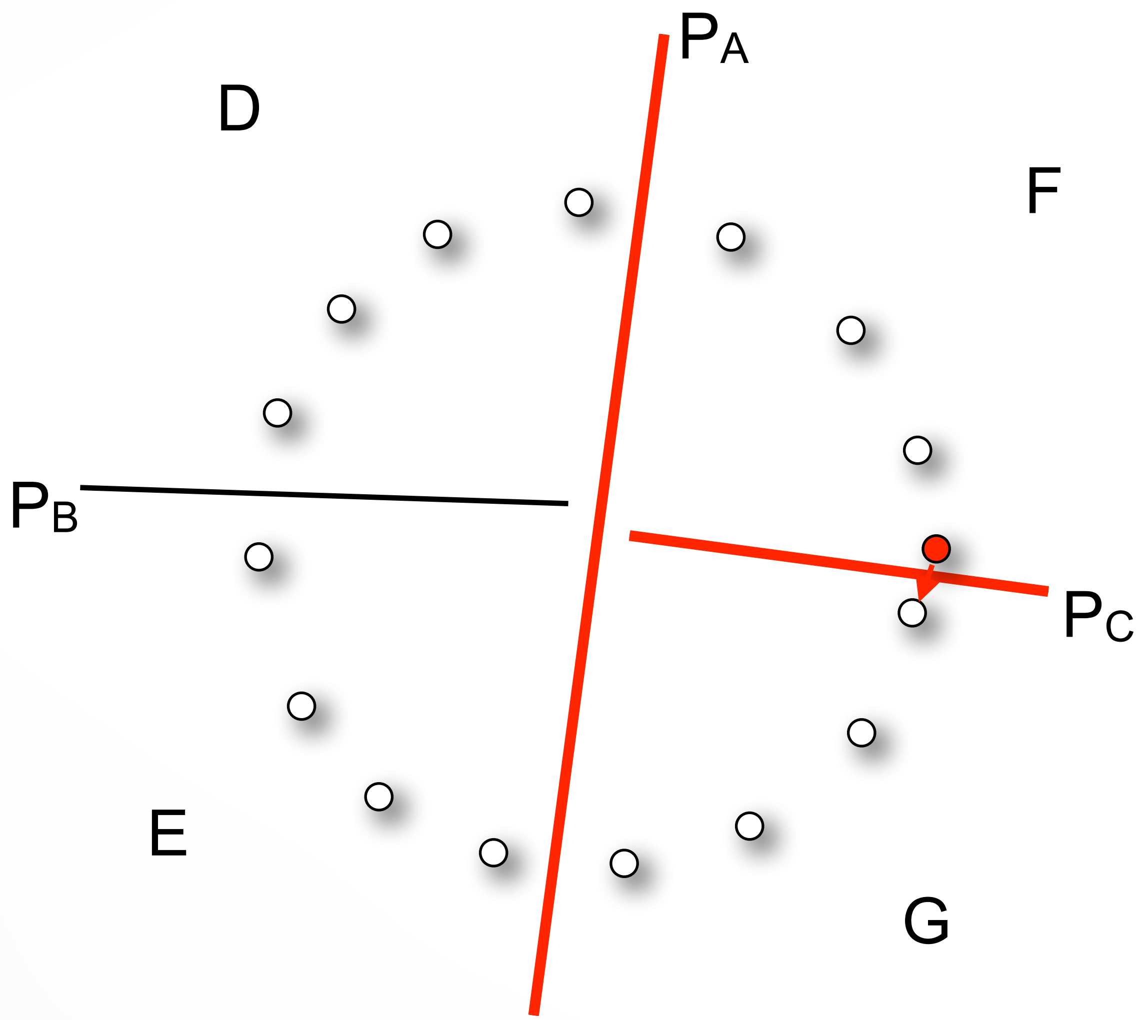


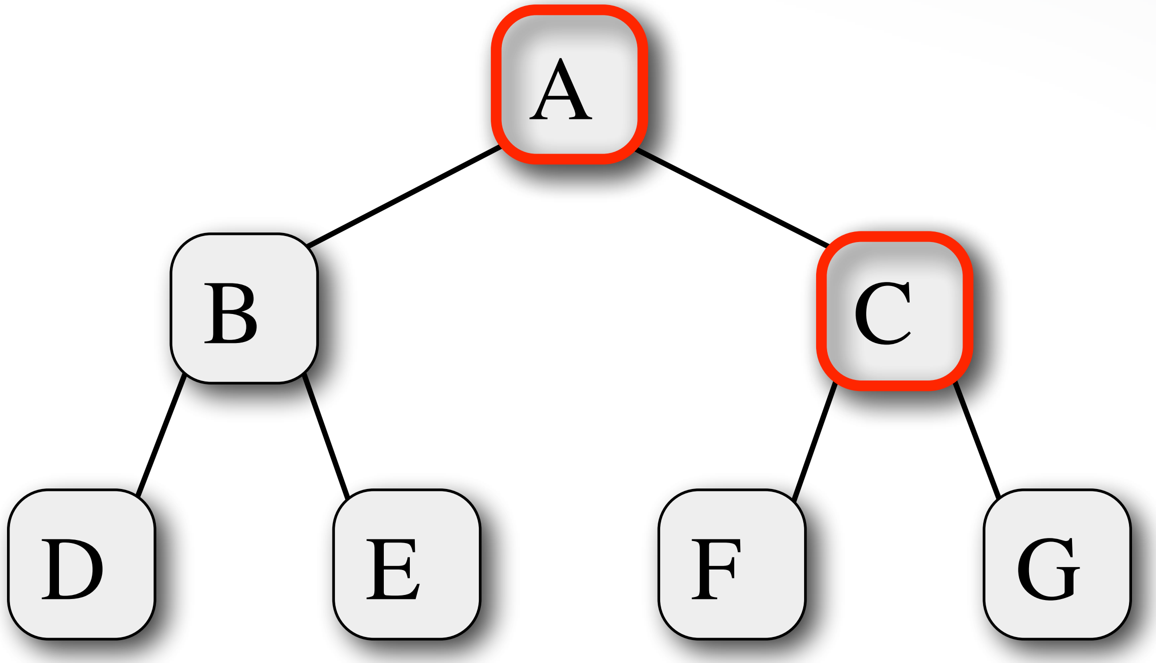
BSP Closest Point Search





BSP Closest Point Search





BSP Closest Point Search

```
BSPNode::dist(Point x, Scalar& dmin)
{
  if (leaf_node())
    for each sample point p[i]
      dmin = min(dmin, dist(x, p[i]));

  else
  {
    d = dist_to_plane(x);
    if (d < 0)
    {
      left_child->dist(x, dmin);
      if (|d| < dmin) right_child->dist(x, dmin);
    }
    else
    {
      right_child->dist(x, dmin);
      if (|d| < dmin) left_child->dist(x, dmin);
    }
  }
}
```

Closest Compatible Point

- Closest point often a bad approximation to corresponding point
- Can improve matching effectiveness by restricting match to **compatible** points
  - Compatibility of colors [Godin et al. ’94]
  - Compatibility of normals [Pulli ’99]
  - Other possibilities: curvature, higher-order derivatives, and other local features (remember: data is noisy)

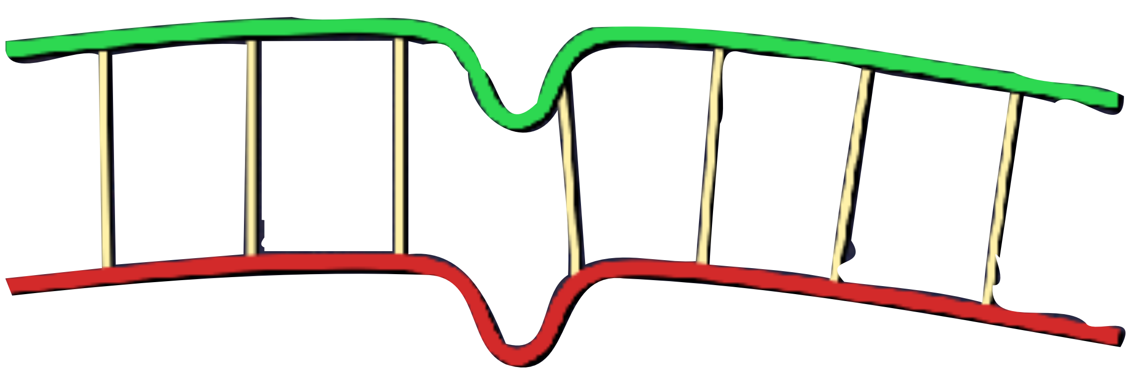
ICP Variants

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

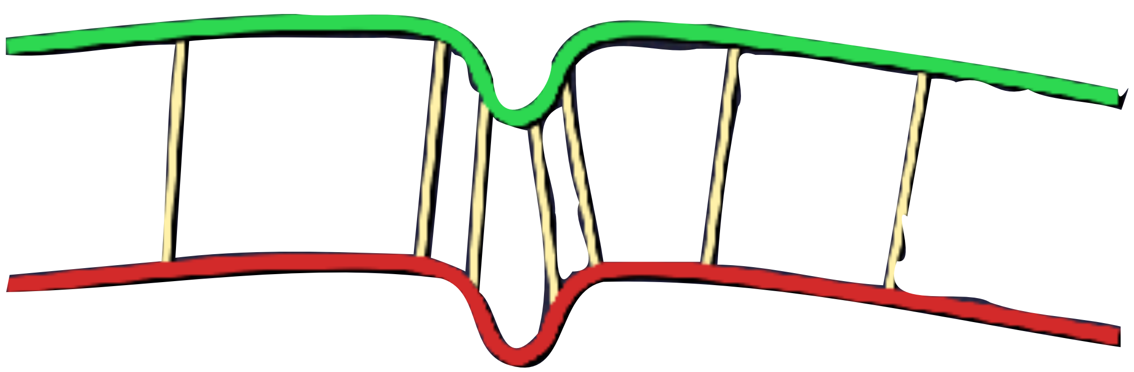
Selecting Source Points

- Use all points
- Uniform subsampling
- Random sampling
- **Stable sampling** [Gelfand et al. 2003]
    - Select samples that constrain all degrees of freedom of the rigid-body transformation

Stable Sampling



Uniform Sampling



Stable Sampling

Covariance Matrix

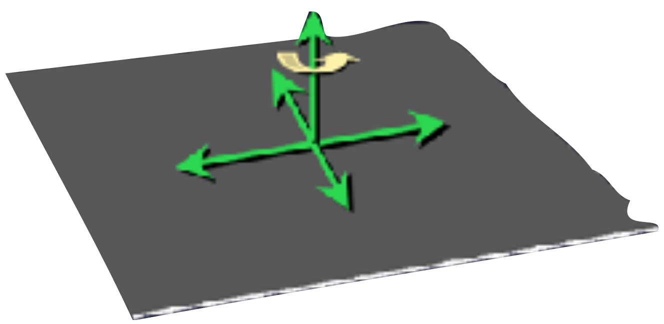
- Aligning transform is given by $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \leftarrow & \mathbf{p}_1 \times \mathbf{n}_1 & \rightarrow & \leftarrow & \mathbf{n}_1 & \rightarrow \\ \leftarrow & \mathbf{p}_2 \times \mathbf{n}_1 & \rightarrow & \leftarrow & \mathbf{n}_2 & \rightarrow \\ & \vdots & & & \vdots & \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ t_x \\ t_y \\ t_z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -(\mathbf{p}_1 - \mathbf{q}_1)^\top \mathbf{n}_1 \\ -(\mathbf{p}_2 - \mathbf{q}_2)^\top \mathbf{n}_2 \\ \vdots \end{bmatrix}$$

- Covariance matrix $\mathbf{C} = \mathbf{A}^\top \mathbf{A}$ determines the change in error when surfaces are moved from optimal alignment

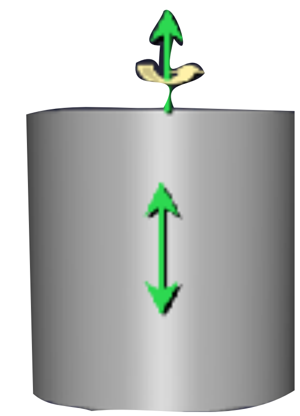
Sliding Directions

- Eigenvectors of $\mathbf{C}$ with small eigenvalues correspond to sliding transformations

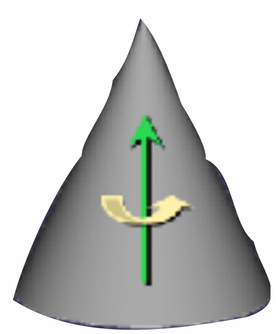


3 small eigenvalues
2 translation
1 rotation

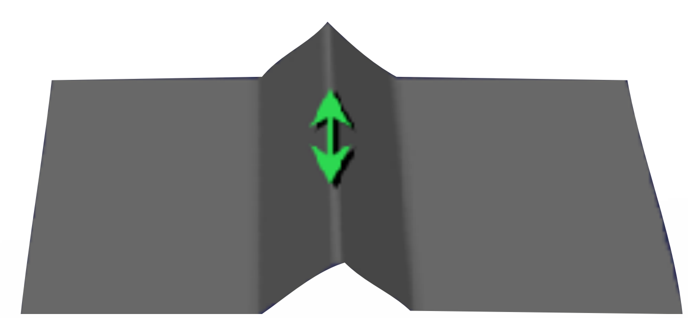
3 small eigenvalues
3 rotation



2 small eigenvalues
1 translation
1 rotation



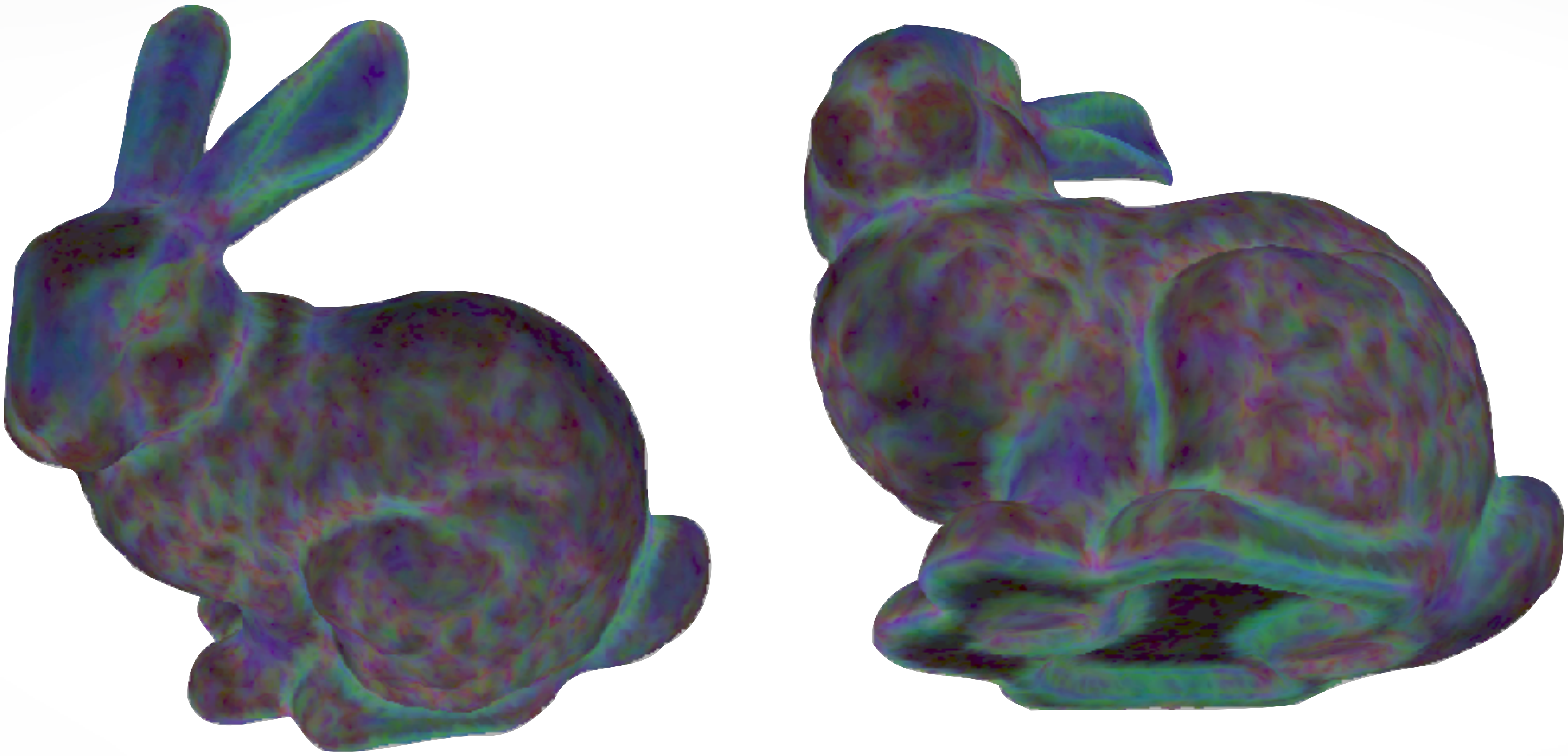
1 small eigenvalue
1 rotation



1 small eigenvalue
1 translation

[Gelfand]

Stability Analysis



Key:

■ (black)	3 DOFs stable	■ (green)	5 DOFs stable
■ (blue)	4 DOFs stable	■ (red)	6 DOFs stable

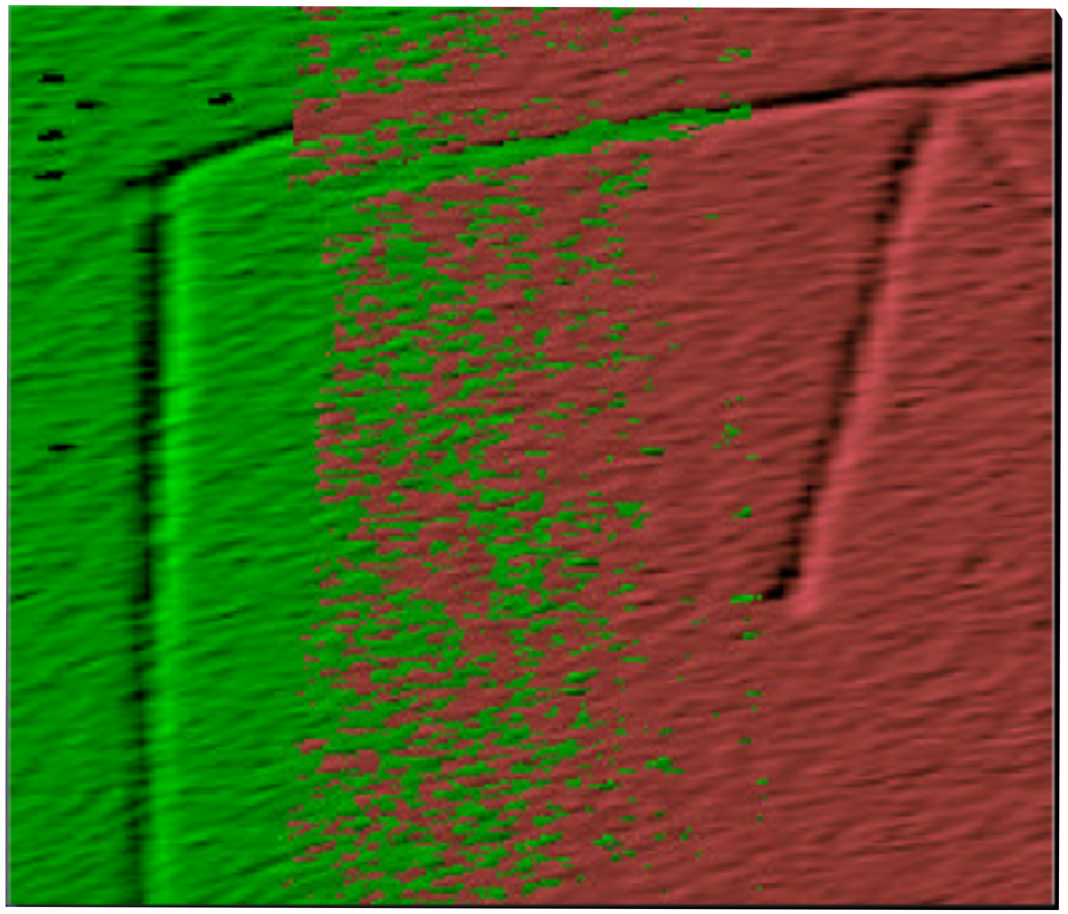
Sample Selection

- Select points to prevent small eigenvalues
  - Based on $\mathbf{C}$ obtained from sparse sampling

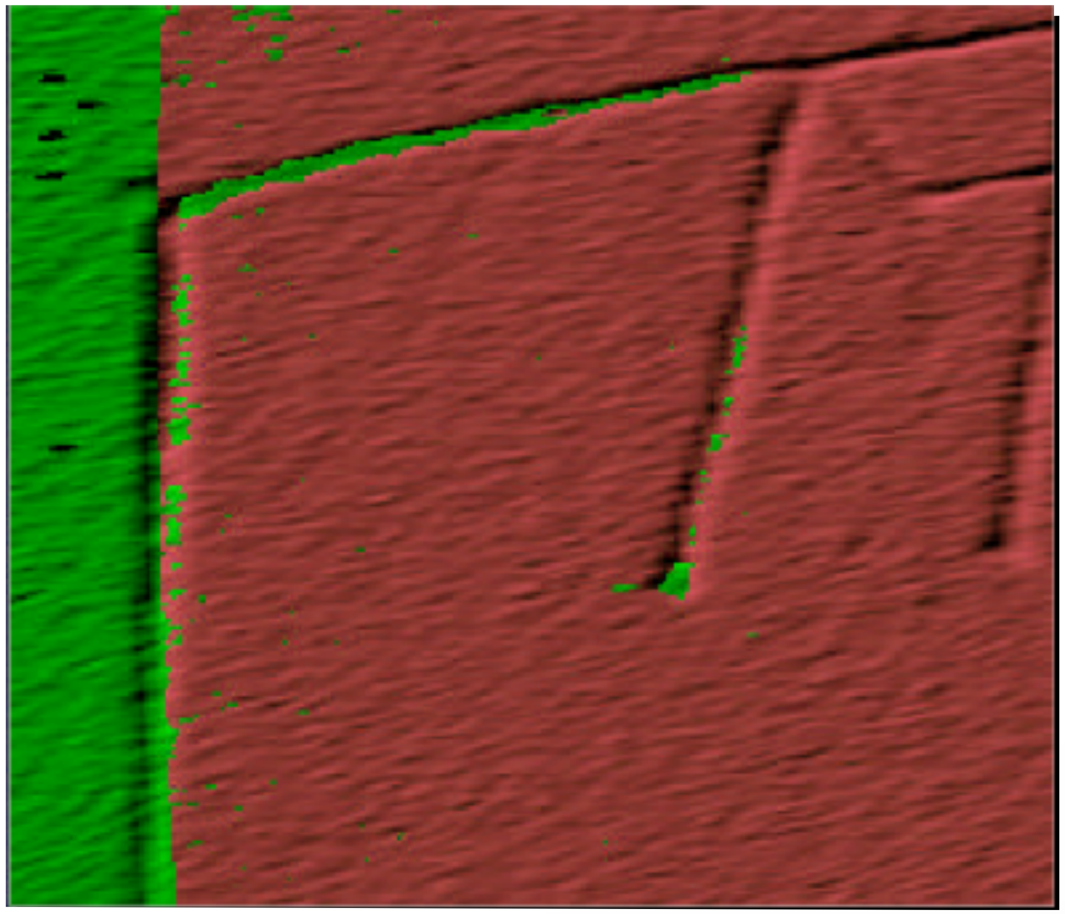
- Simpler variant: normal-space sampling
  - select points with uniform distribution of normals
  - **Pro**: faster, does not require eigenanalysis
  - **Con**: only constrains translation

Result

Stability-based or normal-space sampling important for smooth areas with small features



Random Sampling



Normal-space Sampling

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to ensure enough samples in features except at high sampling rates
- However, have to build special data structure
- Preprocessing / run-time cost tradeoff

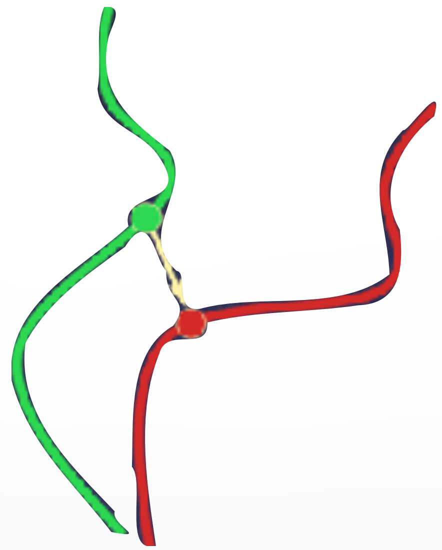
Improving ICP Speed

Projection-based matching

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

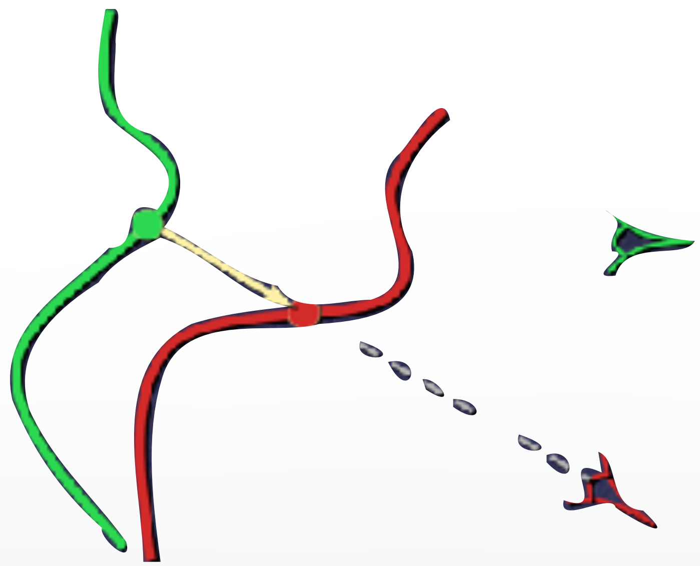
Finding Corresponding Points

- Finding Closest point is most expensive stage of the ICP algorithm
  - Brute force search – O(n)
  - Spatial data structure (e.g., k-d tree) – O(log n)



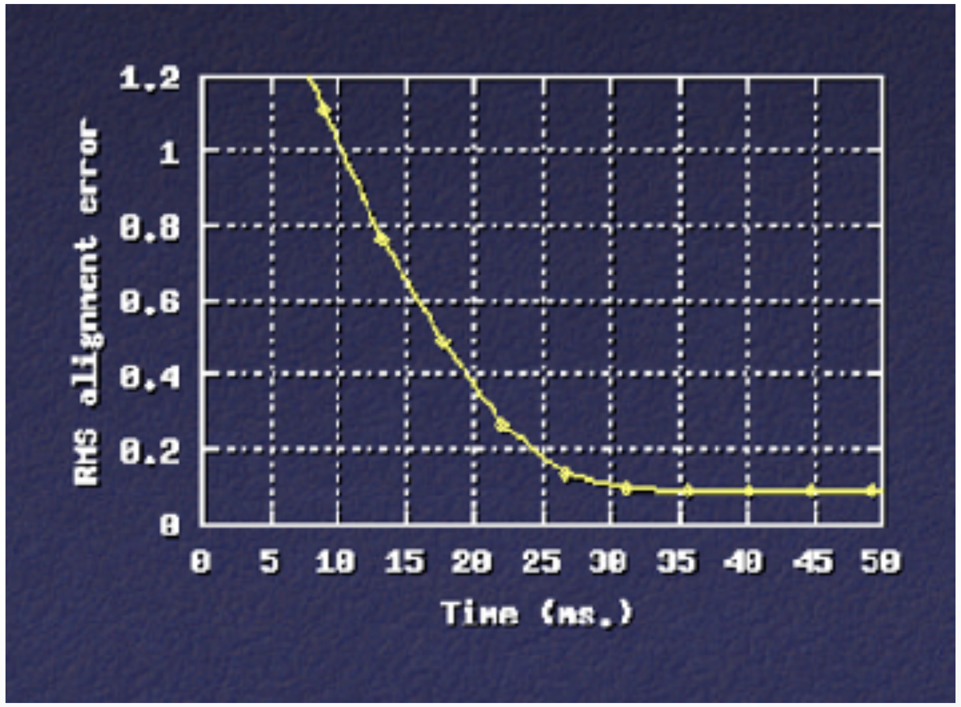
Projection to Find Correspondence

- Idea: use a simpler algorithm to find correspondences
- For range images, can simply project point [Blais 95]
  - Constant-time
  - Does not require precomputing a spatial data structure



Projection-Based Matching

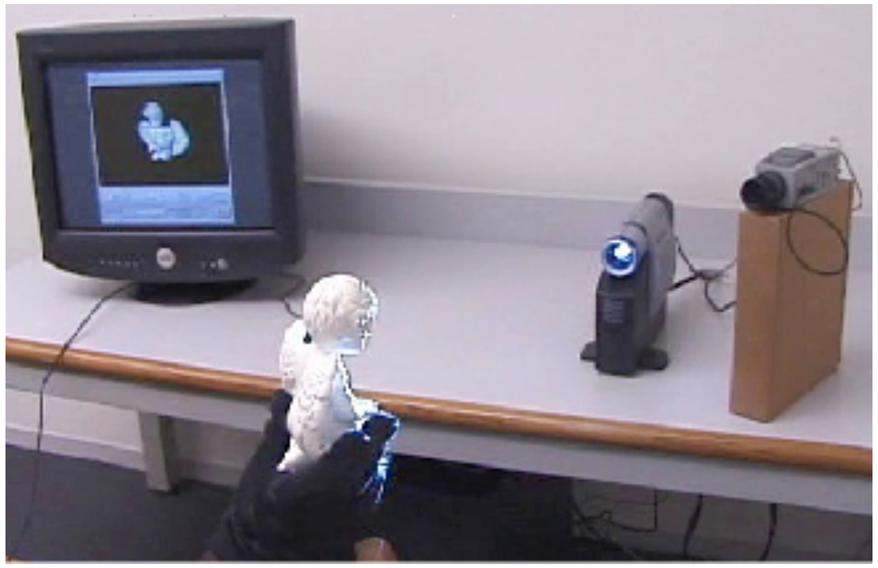
- Slightly worse performance per iteration
- Each iteration is one to two orders of magnitude faster than closest point
- Result: can align two range images in a few milliseconds, vs. a few seconds



Application

- Given:
  - A scanner that returns range images in real time
  - Fast ICP
  - Real-time merging and rendering
- Result: 3D model acquisition
  - Tight feedback loop with user
  - Can see and fill holes while scanning

Examples



[Rusinkiewicz et al. ‘02]

Artec Group

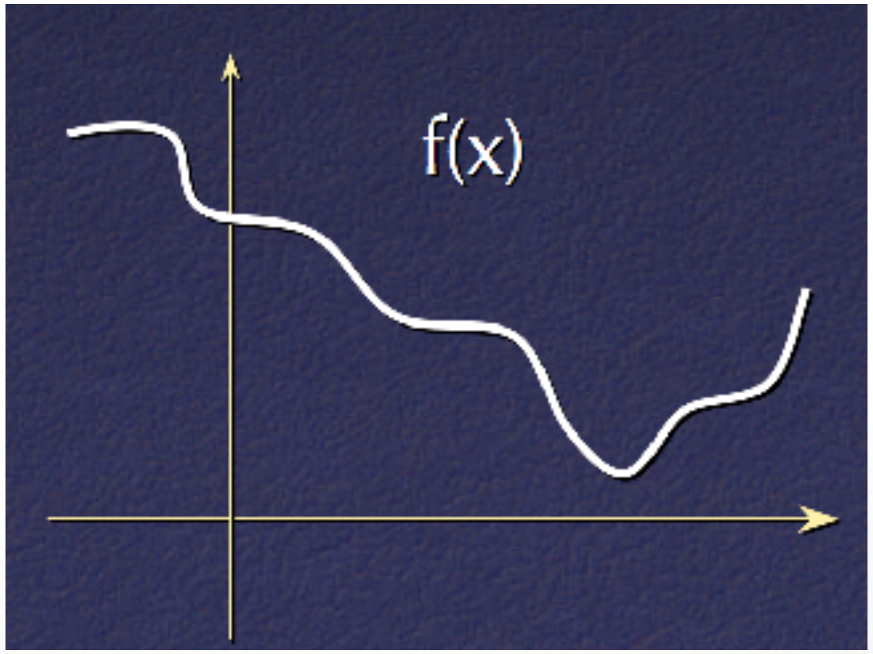
[Newcombe et al. ’11]
KinectFusion

Theoretical Analysis of ICP Variants

- One way of studying performance is via empirical tests on various scenes
- How to analyze performance analytically?
- For example, when does point-to-plane help? Under what conditions does projection-based matching work?

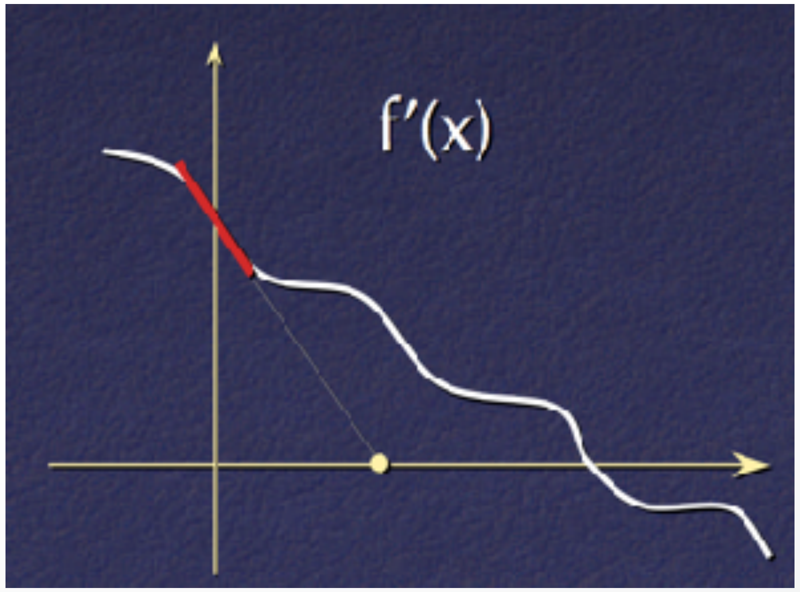
What does ICP do?

- Two ways of thinking about ICP:
  - Solving correspondence problem
  - **Minimizing point-to-surface squared distance**
- ICP is like Newton's method on an approximation of the distance function



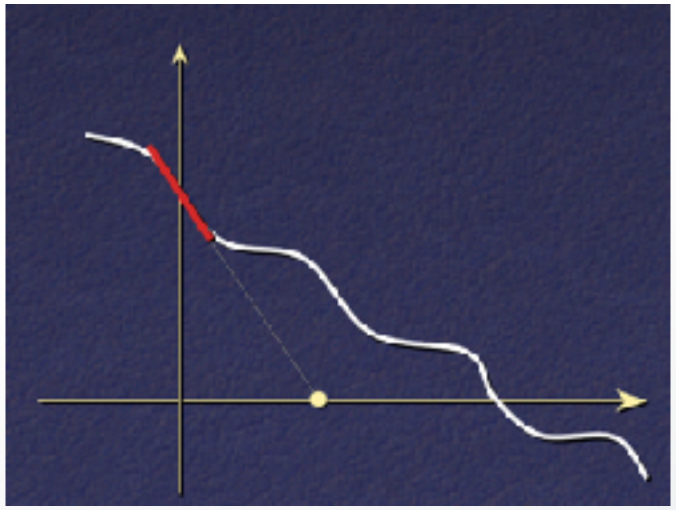
What does ICP do?

- Two ways of thinking about ICP:
  - Solving correspondence problem
  - **Minimizing point-to-surface squared distance**
- ICP is like Newton's method on an approximation of the distance function

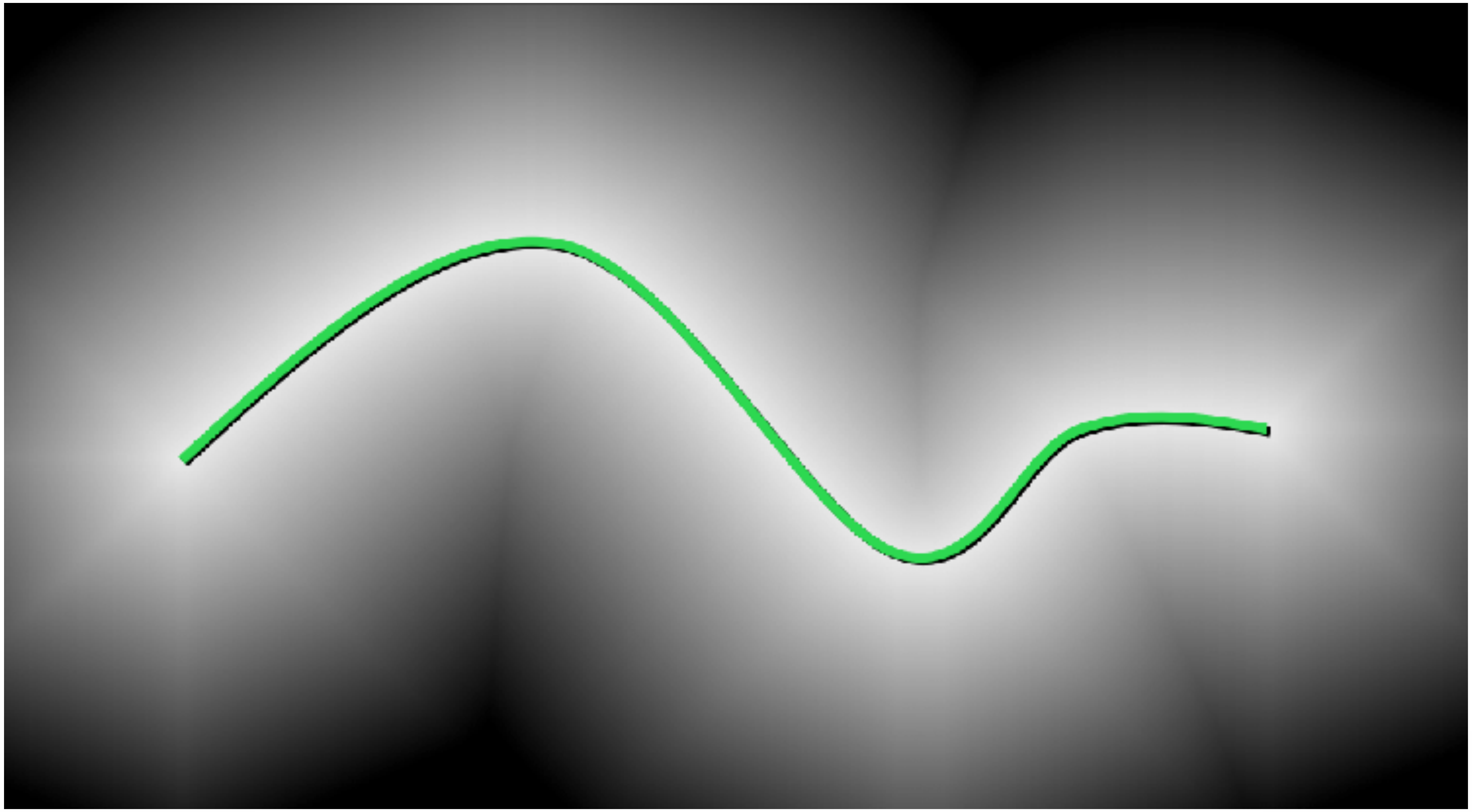


What does ICP do?

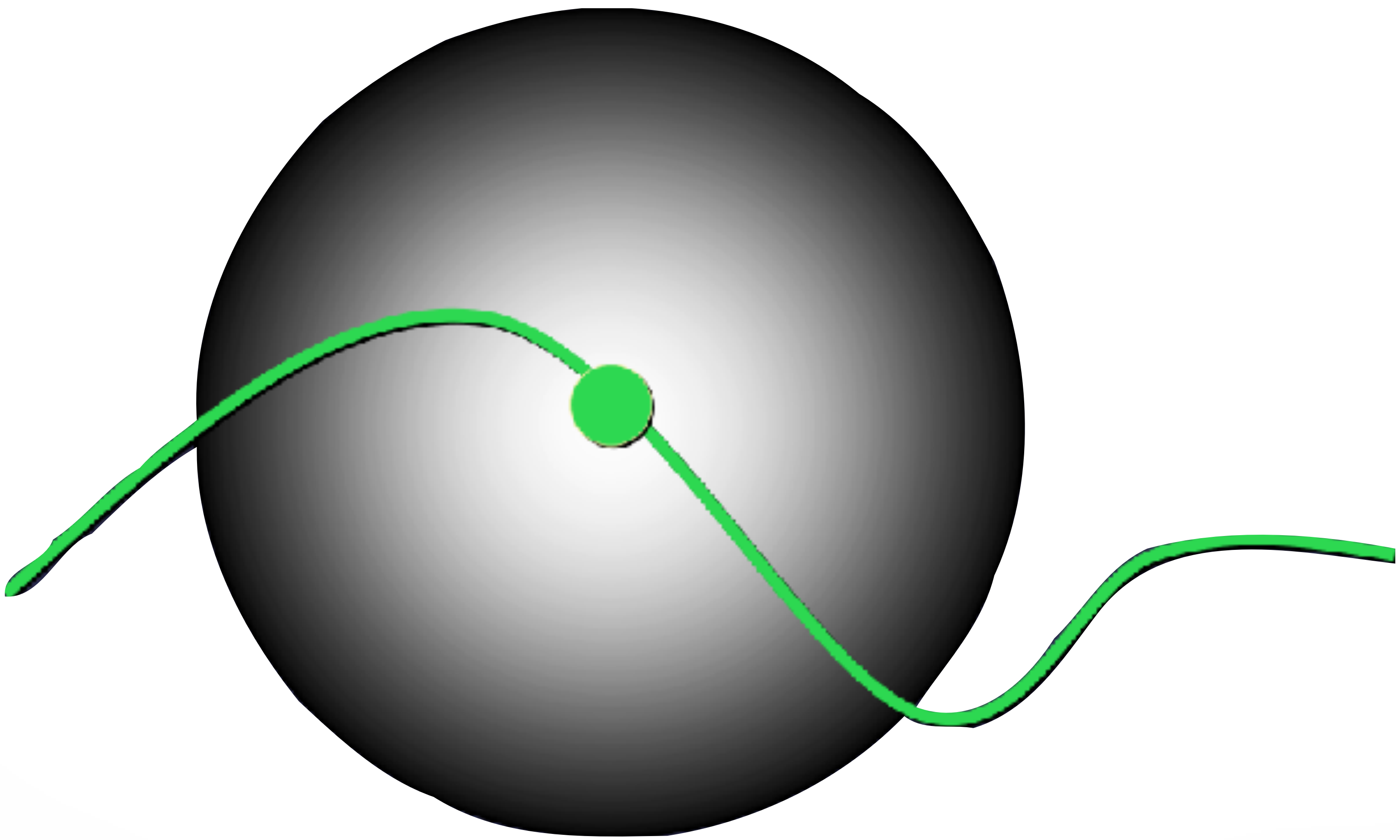
- Two ways of thinking about ICP:
  - Solving correspondence problem
  - **Minimizing point-to-surface squared distance**
- ICP is like Newton's method on an approximation of the distance function
  - ICP variants affect shape of the global error function or local approximation



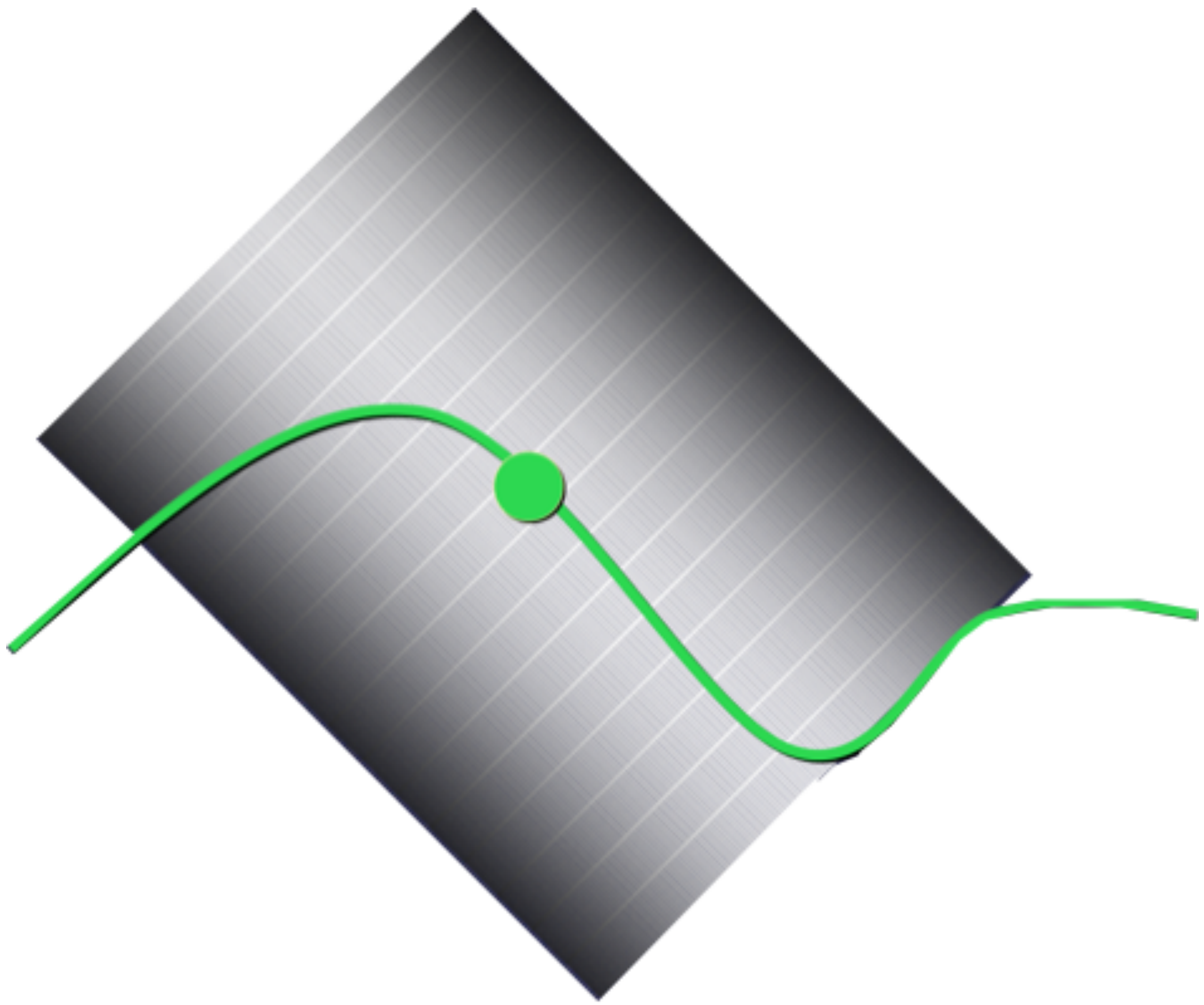
Point-to-Surface Distance



Point-to-Point Distance

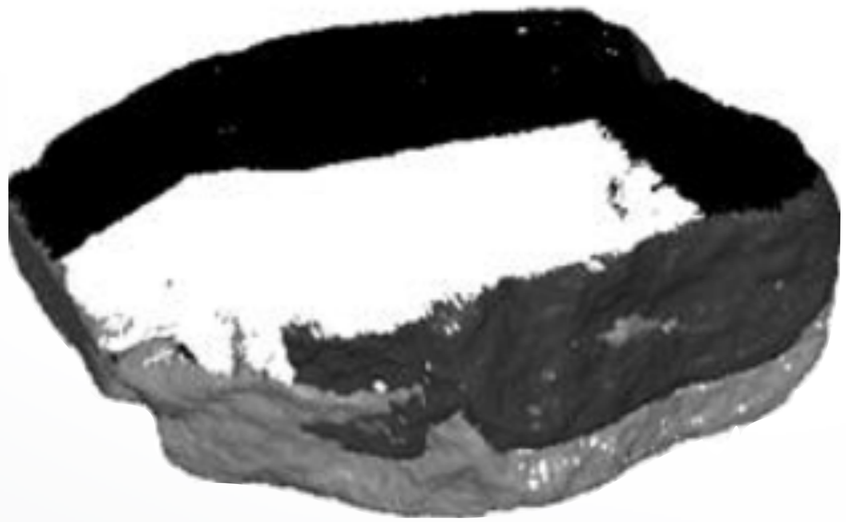


Point-to-Plane Distance



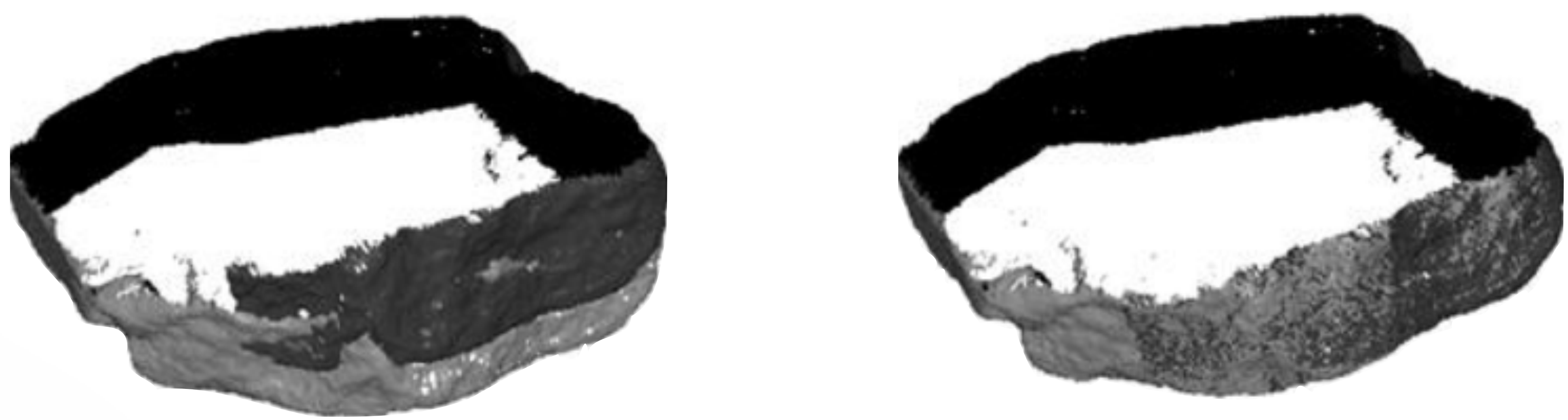
Global Registration Goal

- Given: n scans around an object
- Goal: align them all
- First attempt: apply ICP to each scan to one other



Global Registration Goal

- Want method for distributing accumulated error among all scans



Approach #1: Avoid the Problem

- In some cases, have 1 (possibly low-resolution) scan that covers most surface
- Align all other scans to this “anchor” [Turk 94]
- Disadvantage: not always practical to obtain anchor scan

Approach #2: The Greedy Solution

- Align each new scan to all previous scans [Masuda ’96]
- Disadvantages:
  - Order dependent
  - Doesn’t spread out error

Approach #3: The Brute-Force Solution

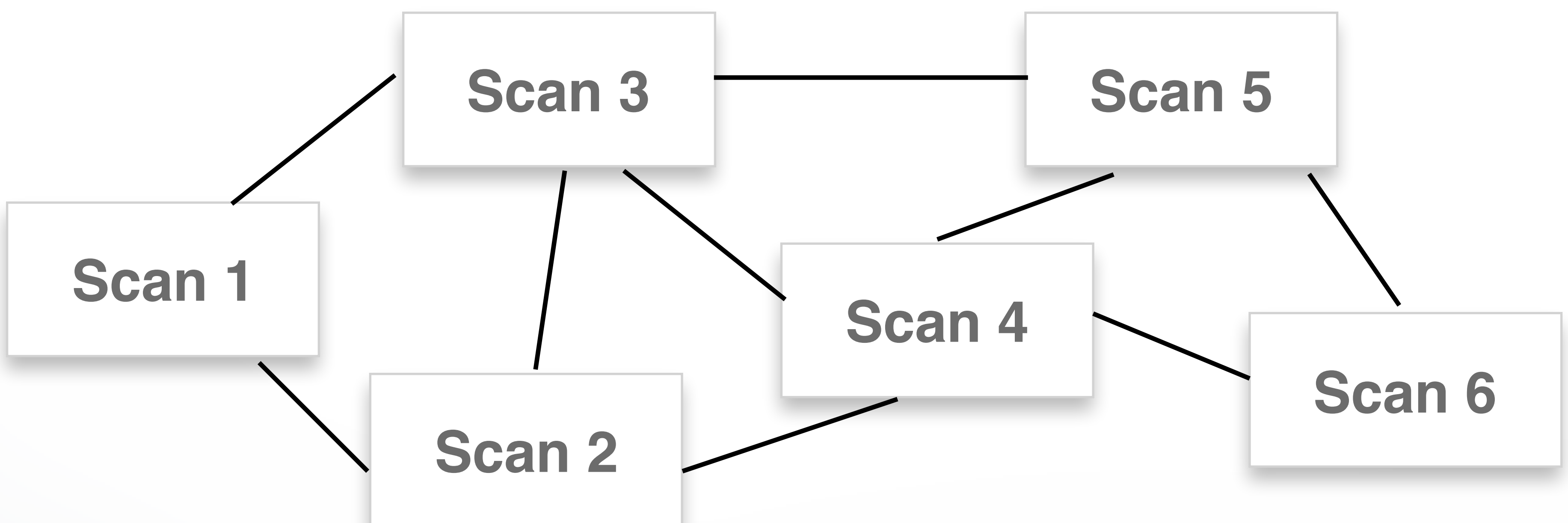
- While not converged:
  - For each scan:
    - For each point:
      - For every other scan
        - Find closest point
  - Minimize error w.r.t. transforms of **all** scans
- Disadvantage:
  - Solve $(6n) \times (6n)$ matrix equation, where n is number of scans

Approach #3a: Slightly Less Brute-Force Solution

- While not converged:
  - For each scan:
    - For each point:
      - For every other scan
        - Find closest point
  - Minimize error w.r.t. transforms of **this** scans
- Faster than previous method (matrices are 6x6) [Bergevin '96, Benjemaa '97]

Graph Methods

- Many global registration algorithms create a graph of **pairwise alignments** between scans



Sharp et al. Algorithm

- Perform pairwise ICPs, record sample (e.g., 200) of corresponding points
- For each scan, starting w most connected
  - Align scan to existing set
  - While (change in error) > threshold
    - Align each scan to others
- All alignments during global reg phase use precomputed corresponding points.

Lu and Milios Algorithm

- Perform pairwise ICPs, record optimal rotation/translation and covariance for each
- Least squares simultaneous minimization of all errors (covariance-weighted)
- Requires linearization of rotations
  - Worse than the ICP case, since don't converge to (incremental rotation) = 0

Bad ICP in Global Registration

One bad ICP can throw off the entire model!

Correct Global Registration

Global Registration Including Bad ICP

Literature

- Rusinkiewicz & Levoy, Efficient Variants of the ICP Algorithm, 3DIM 2001
- Chen & Medioni, “Object modeling by registration of multiple range images”, ICRA1991
- Besl & McKay: A method for registration of 3D shapes, PAMI 1992
- Horn: Closed-form solution of absolute orientation using unit quaternions, Journal Opt. Soc. Amer. 4(4), 1987
- Gelfand et al: Geometrically Stable Sampling for the ICP Algorithm, 3DIM, 2001.
- Pulli, Multiview Registration for Large Data Sets, 3DIM 1999

http://cs621.hao-li.com

Thanks!