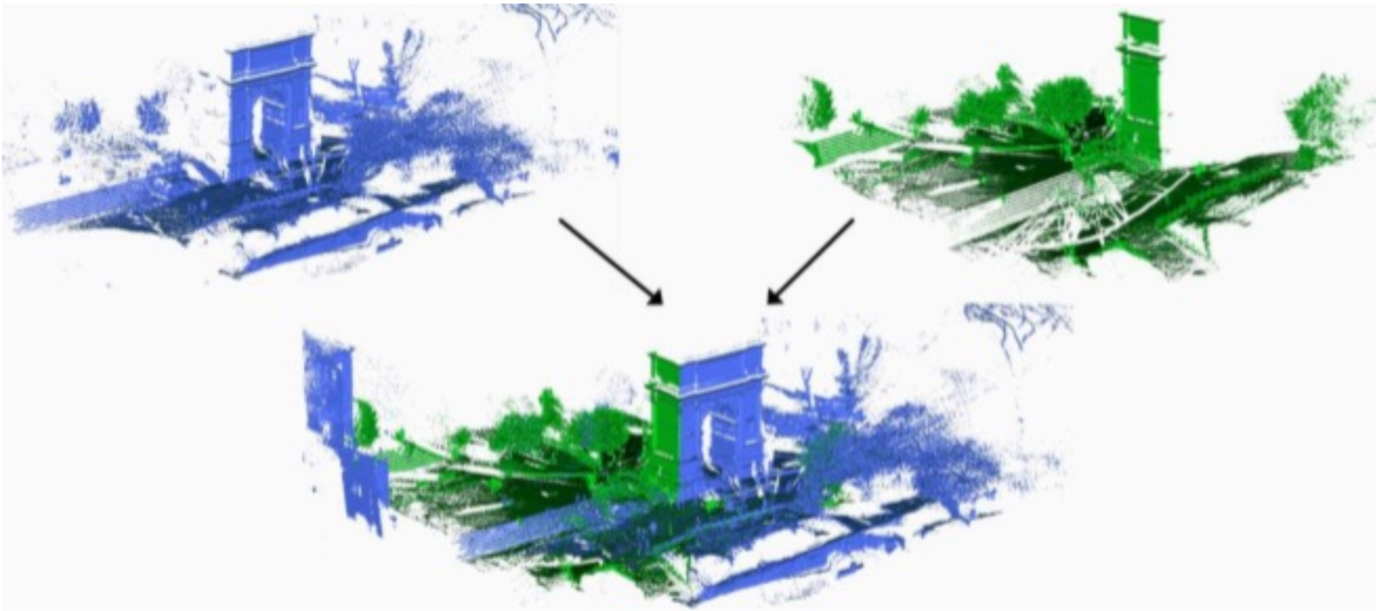
中国科学技术大学
University of Science and Technology of China

Geometry Registration

张举勇
中国科学技术大学

什么是注册？



- 计算最佳空间变换，以使得多个几何曲面之间进行对齐。
  - 将传感器采集的多个局部测量数据拼接成一个完整的几何模型
  - 将新测量数据对齐到已知模型以估计其姿态

变换类型

- Same object in a different position: size and shape preserving
  - Rigid-body transformation (rotation and translation)
  - Six degrees of freedom
    - ✦ translation $\mathbf{t} = (t_x, t_y, t_z)^T$
    - ✦ rotation (α, β, γ)

$$\mathbf{T}_{\text{rigid}}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t}$$

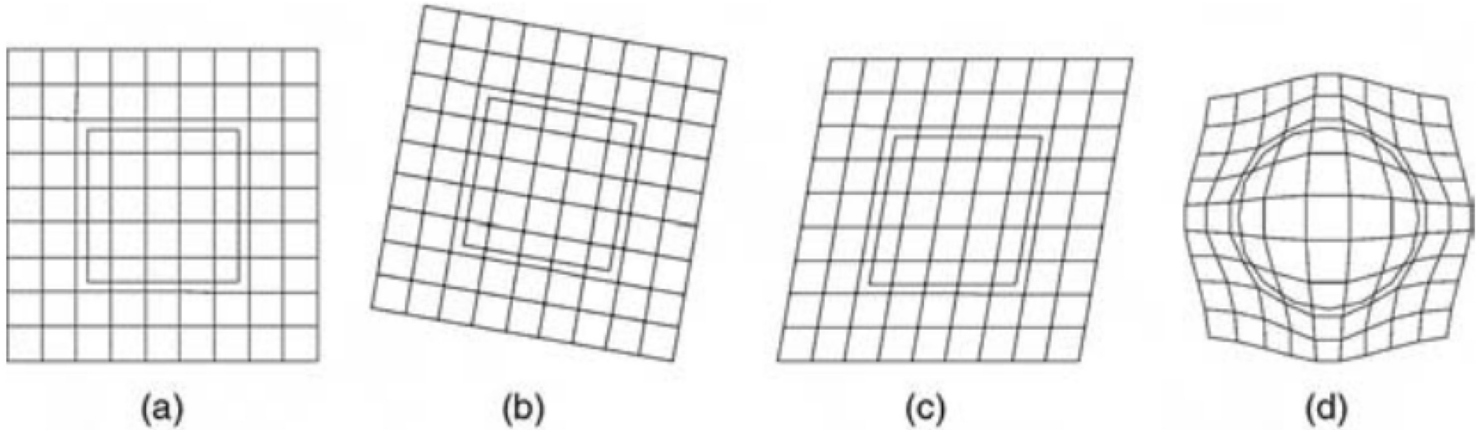
$$\mathbf{T}_{\text{rigid}}(\mathbf{x}) = \begin{bmatrix} \cos\beta\cos\gamma & \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & t_x \\ -\cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & t_y \\ \sin\beta & -\sin\alpha\cos\beta & \cos\alpha\cos\beta & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

变换类型

- Affine or Linear Transformation
  - Rigid-body transformation (rotation and translation)
  - Scaling and Shearing
  - Twelve degrees of freedom

$$\mathbf{T}(\mathbf{x}) = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

变换类型



Example of different types of transformations of a square

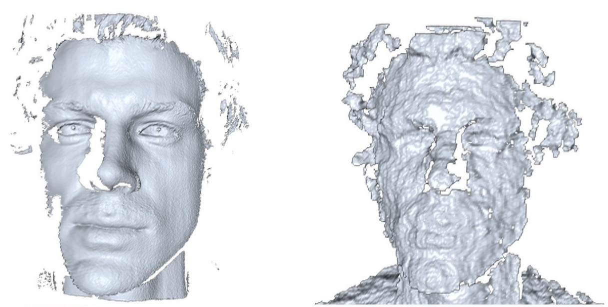
(a) identity transformation

(b) rigid transformation

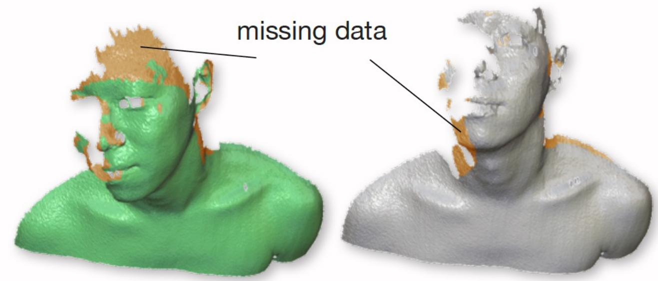
(c) affine transformation

(d) nonrigid transformation

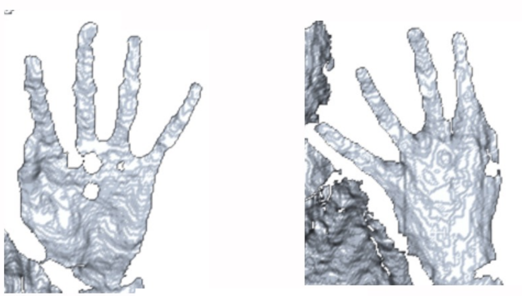
配准问题中的一些挑战



Noise



Partial matching



Ambiguity

Illumination changes

配准问题建模

- 将配准问题表达为能量最小化问题：

$$\underset{T}{\operatorname{argmax}} E_{reg}(T,P,Q)$$

$$E_{reg}(T,P,Q) = E_{match}(T,P,Q) + E_{prior}(T)$$

配准误差

如何衡量配准结果的质量？

变换误差

变换的类型与表示方式？

配准问题建模

- 配准误差

$$E_{reg}(T,P,Q) = E_{match}(T,P,Q) + E_{prior}(T)$$

$$E_{match}(T,P,Q) = \int_X \phi(T(p),Q)dx$$

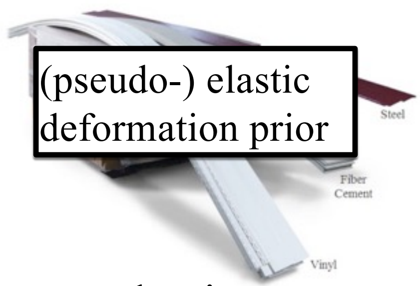
距离度量函数

配准问题建模

- 变换误差

$$E_{reg}(T,P,Q) = E_{match}(T,P,Q) + E_{prior}(T)$$

Rigid



Elastic

Articulated

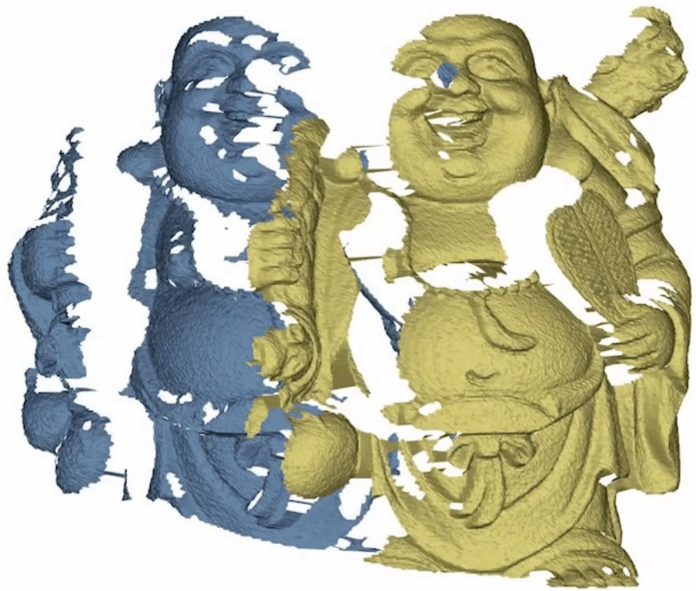
Composite

几何数据融合与跟踪-刚性注册

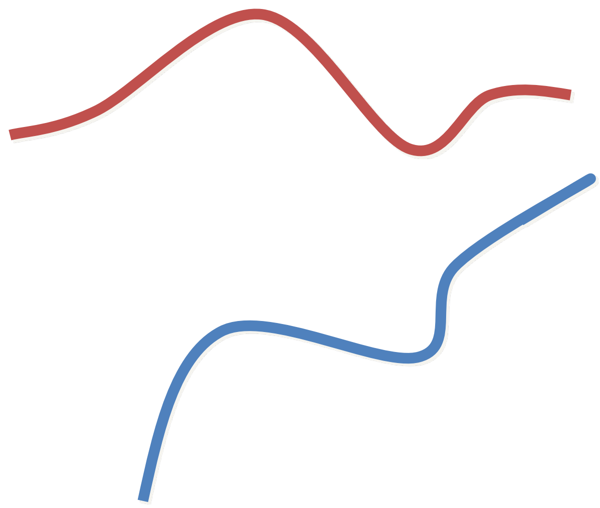
- 刚体几何建模：将不同视角点云进行刚性拼接，以获得完整几何模型

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in K} \| \mathbf{p} - \mathbf{T}\mathbf{q} \|^2$$

T 是一个包含旋转与平移的刚性变换



Aligning 3D Data

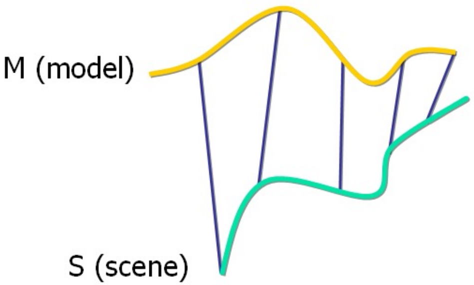


Corresponding Point Set Alignment

- Let M be a model point set.
- Let S be a scene point set.

We assume:

1. $N_M = N_S$.
2. Each point S_i correspond to



Corresponding Point Set Alignment

The MSE objective function :

$$f(R,T) = \frac{1}{N_S} \sum_{i=1}^{N_S} \left\| m_i - Rot(s_i) - Trans \right\|^2$$

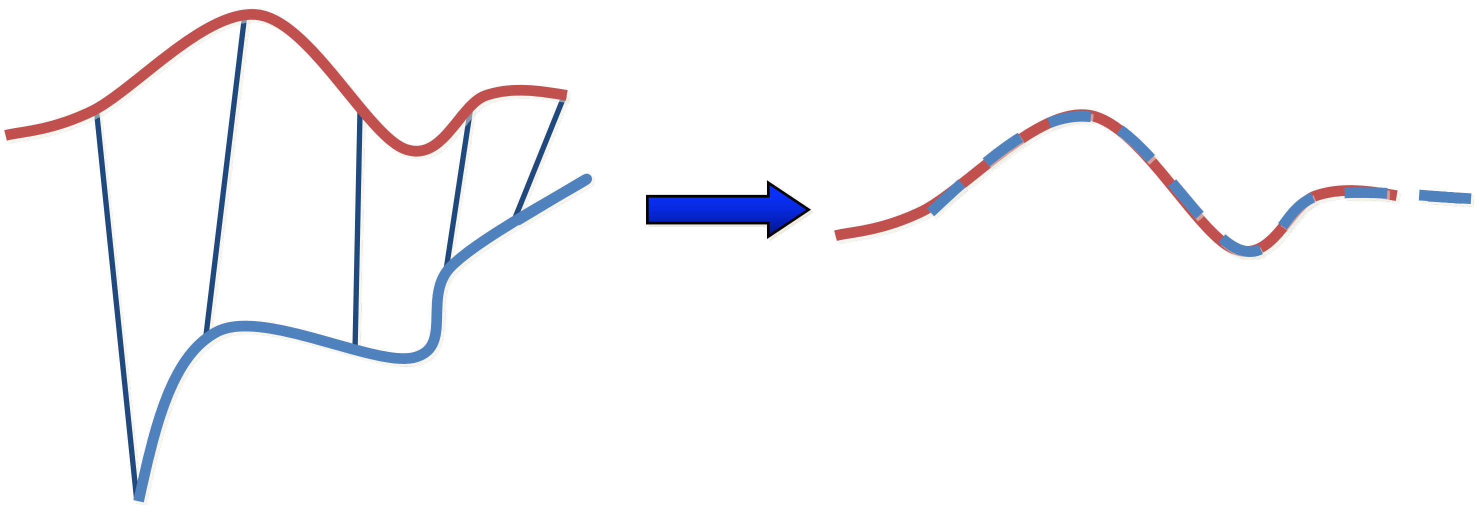
$$f(q) = \frac{1}{N_S} \sum_{i=1}^{N_S} \left\| m_i - R(q_R) s_i - q_T \right\|^2$$

The alignment is:

$$(rot, trans, d_{mse}) = \Phi(M, S)$$

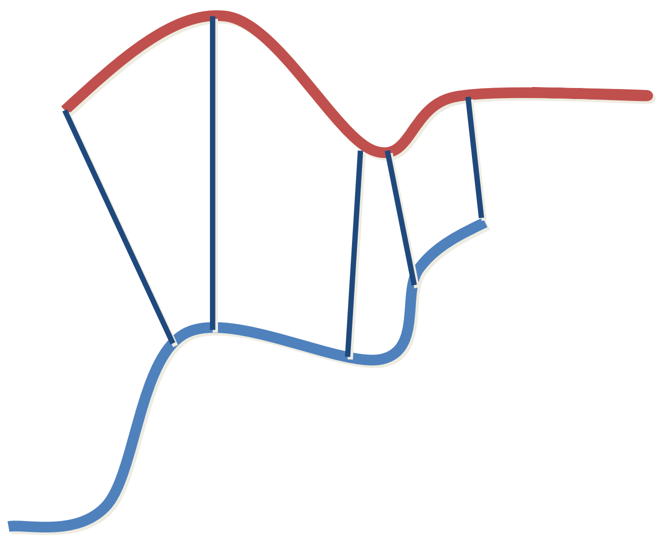
Aligning 3D Data

- If correct correspondences are known, can find correct relative rotation/translation



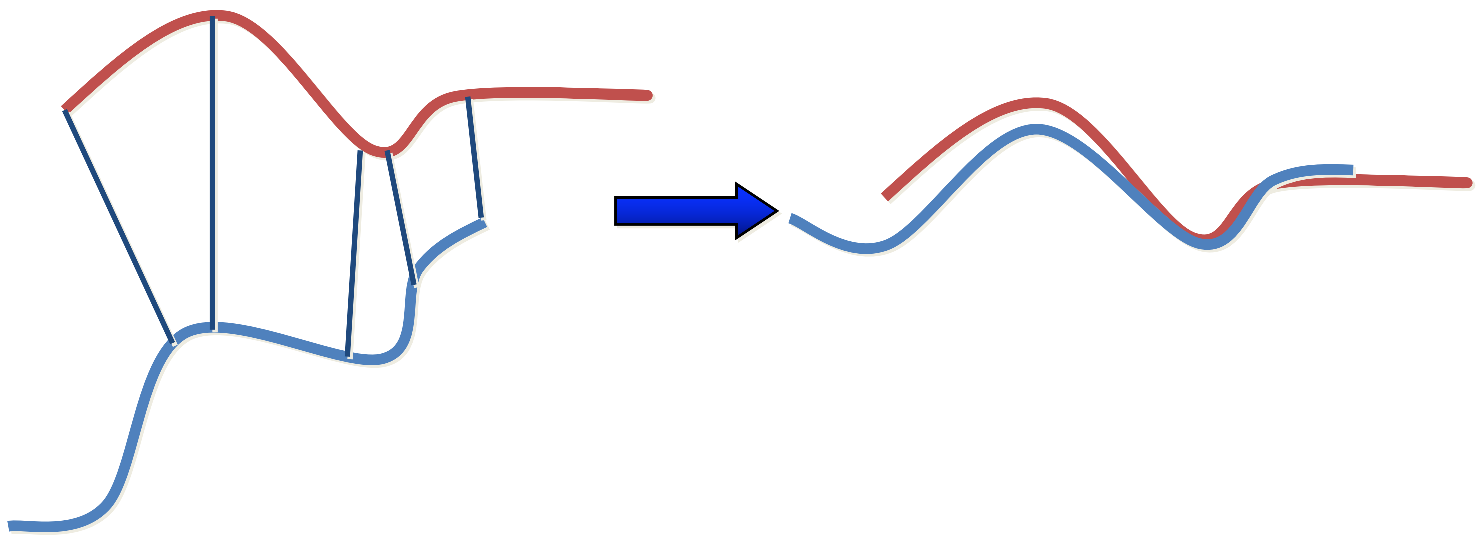
Aligning 3D Data

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



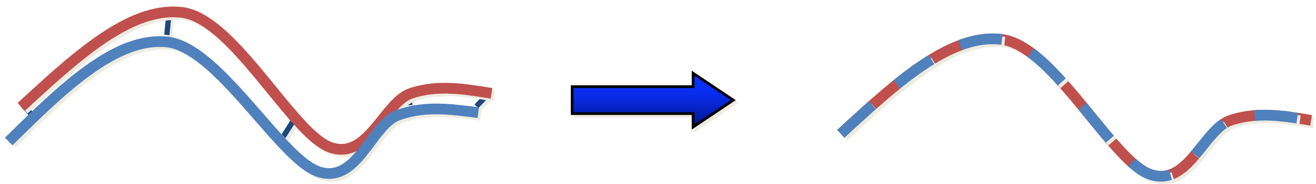
Aligning 3D Data

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



Aligning 3D Data

- Converges if starting position “close enough“



Closest Point

- Given 2 points r_1 and r_2 , the Euclidean distance is:

$$d(r_1, r_2) = \|r_1 - r_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- Given a point r_1 and set of points A , the Euclidean distance is:

$$d(r_1, A) = \min_{i \in 1..n} d(r_1, a_i)$$

Finding Matches

- The scene shape S is aligned to be in the best alignment with the model shape M.
- The distance of each point s of the scene from the model is :

$$d(s, M) = \min_{m \in M} d\|m - s\|$$

Finding Matches

$$d(s, M) = \min_{m \in M} d\|m - s\| = d(s, y)$$

$$y \in M$$

$$Y = C(S, M)$$

$$Y \subseteq M$$

C – the closest point operator

Y – the set of closest points to S

Finding Matches

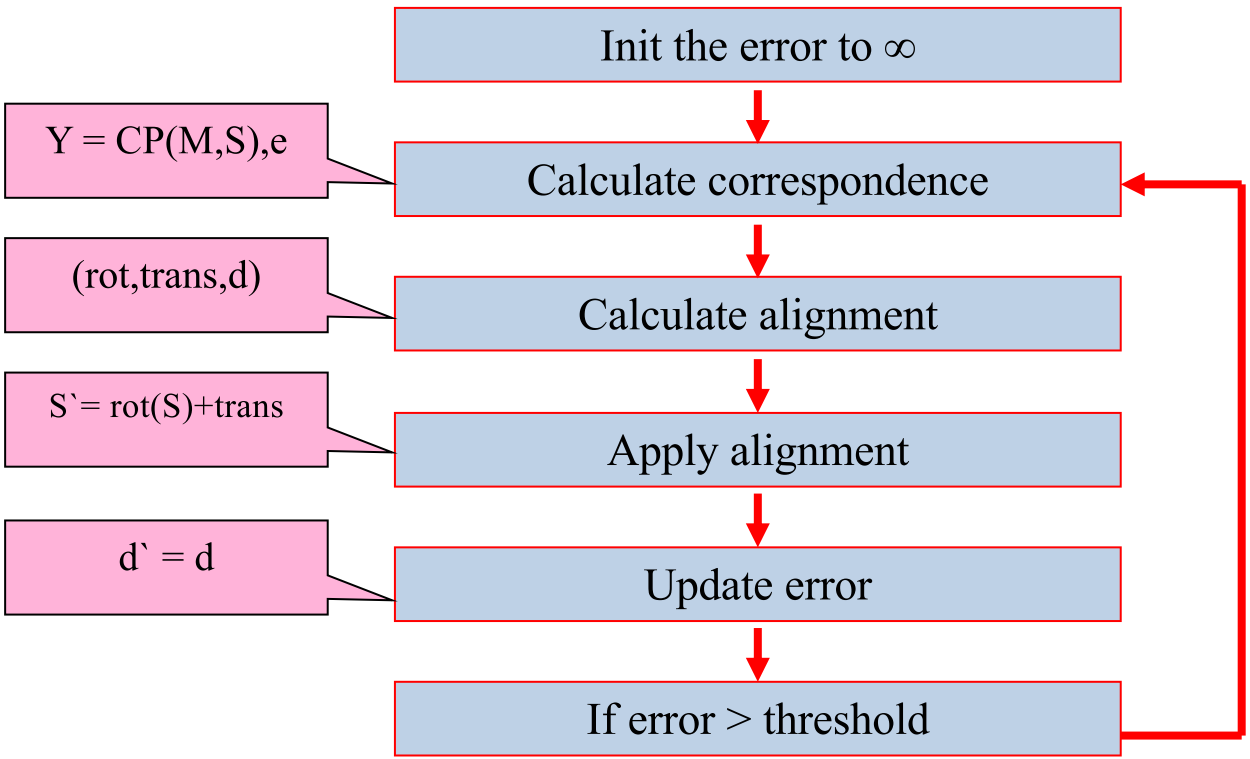
- Finding each match is performed in $O(N_M)$ worst case.
- Given Y we can calculate alignment

$$(rot, trans, d) = \Phi(S, Y)$$

- S is updated to be :

$$S_{new} = rot(S) + trans$$

The Algorithm



Convergence Theorem

- The ICP algorithm always converges monotonically to a local minimum with respect to the MSE distance objective function.

Convergence Theorem

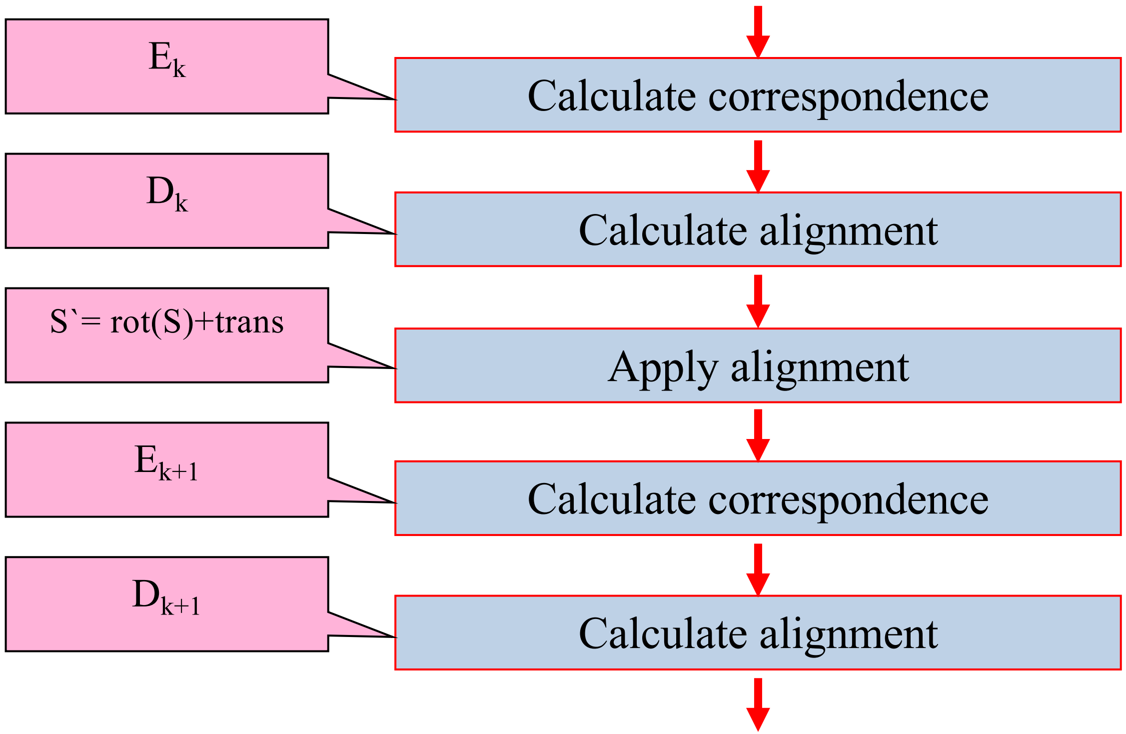
- Correspondence error :

$$e_k = \frac{1}{N_S}\sum_{i=1}^{N_S}\left\| y_{ik} - s_{ik} \right\|^2$$

- Alignment error:

$$d_k = \frac{1}{N_S}\sum_{i=1}^{N_S}\left\| y_{ik} - Rot_k(s_{io}) - Trans_k \right\|^2$$

Convergence Theorem



Convergence Theorem

- Proof :

$$S_k = Rot_k(S_0) + Trans_k$$

$$Y_k = C(M, s_k)$$

$$e_k = \frac{1}{N_S}\sum_{i=1}^{N_S}\left\|y_{ik} - s_{ik}\right\|^2$$

$$d_k = \frac{1}{N_S}\sum_{i=1}^{N_S}\left\|y_{ik} - Rot_k(s_{io}) - Trans_k\right\|^2$$

Convergence Theorem

- Proof : $d_k \le e_k$

  If not - the identity transform would yield a smaller MSE than the least square alignment.

  Apply the alignment q_k on $S_0 \rightarrow S_{k+1}$.

  Assuming the correspondences are maintained : the MSE is still d_k.

$$d_k = \frac{1}{N_M} \sum_{i=1}^{N_M} \left\| y_{ik} - S_{ik} \right\|^2$$

Convergence Theorem

- Proof :

After the last alignment, the closest point operator is applied : $Y_{k+1} = C(M, S_{k+1})$

It is clear that:

$$\left\| y_{i,k+1} - S_{i,k+1} \right\| \leq \left\| y_{ik} - S_{i,k+1} \right\|$$

$$e_{k+1} \leq d_k$$

Thus : $$0 \leq d_{k+1} \leq e_{k+1} \leq d_k \leq e_k$$

Time analysis

Each iteration includes 3 main steps

A. Finding the closest points :

$O(N_M)$ per each point

$O(N_M*N_S)$ total.

B. Calculating the alignment: $O(N_S)$

C. Updating the scene: $O(N_S)$

Optimizing the Algorithm

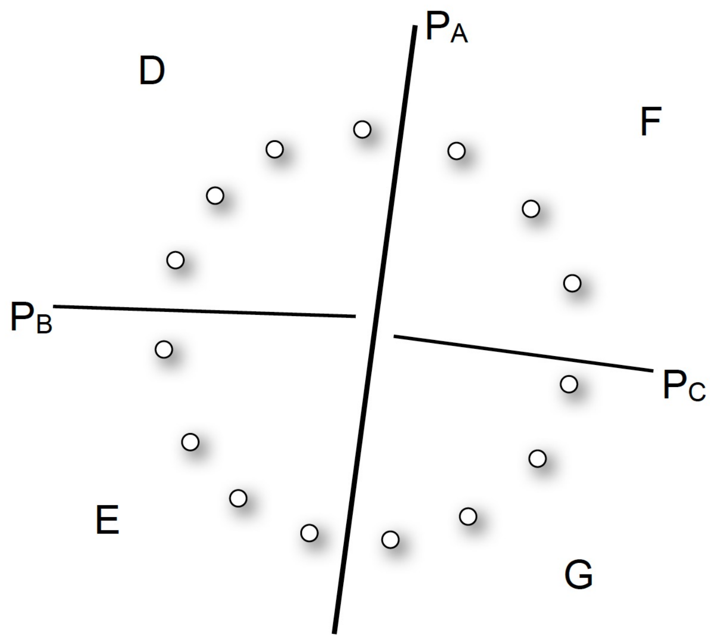
The best match/nearest neighbor problem :

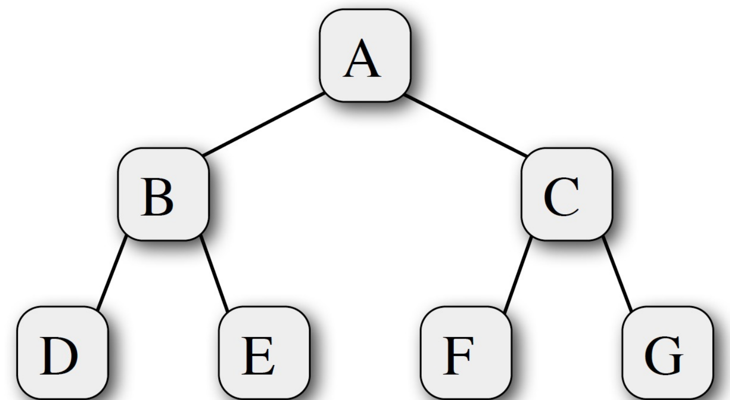
Given a record, and a dissimilarity measure **D**, find the closest record from a set to the query record.

Closest Point Search

- Find closest point of a query point
  - Brute force: O(n) complexity
- Use hierarchical BSP tree
  - Binary space partitioning tree (also kD-tree)
  - Recursively partition 3D space by planes
  - Tree should be balanced, put plane at median
  - log(n) tree levels, complexity O(log n)

BSP Closest Point

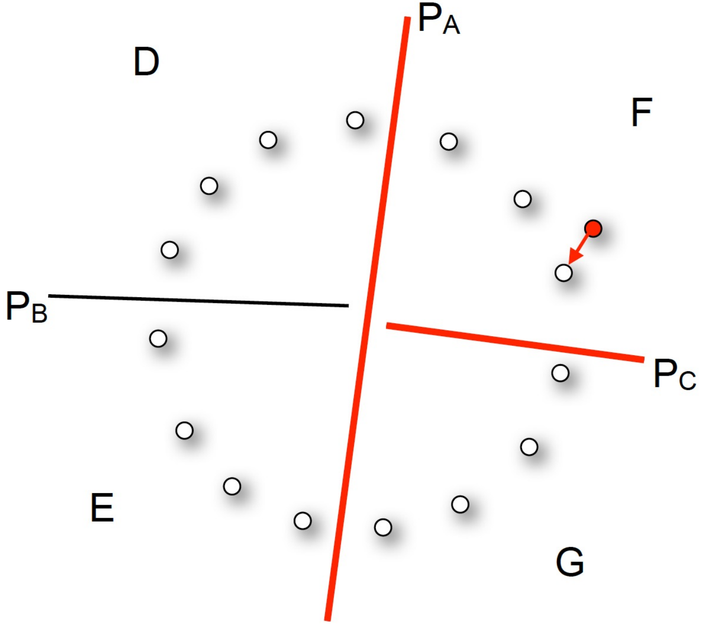


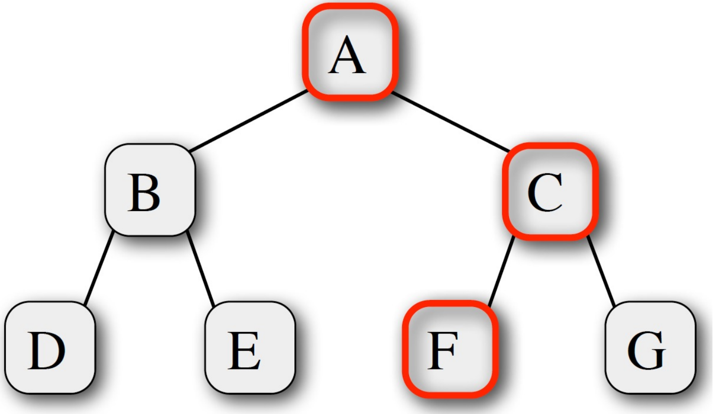


BSP Closest Point

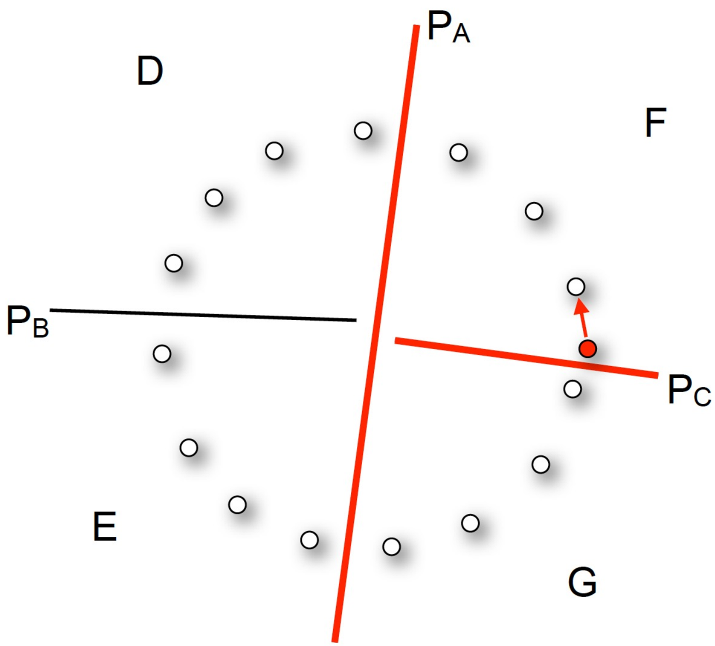
```
BSPNode::dist(Point x, Scalar& dmin)
{
  if (leaf_node())
    for each sample point p[i]
      dmin = min(dmin, dist(x, p[i]));

  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);
    }
  }
}
```

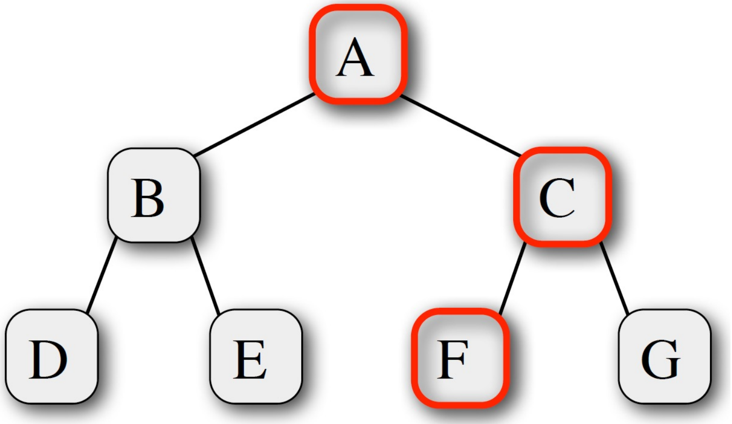
BSP Closest Point



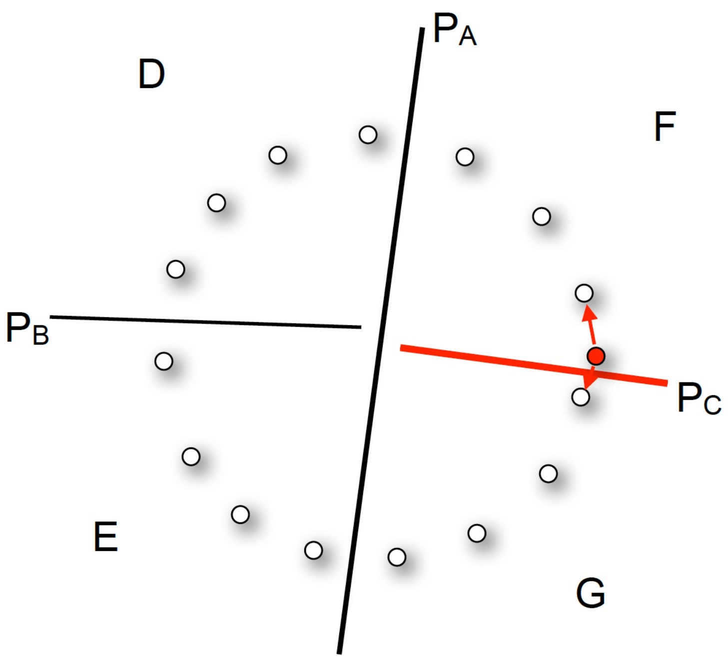


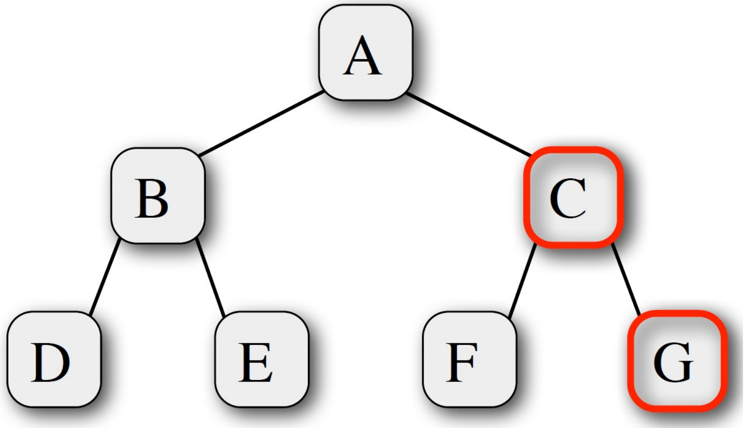
BSP Closest Point



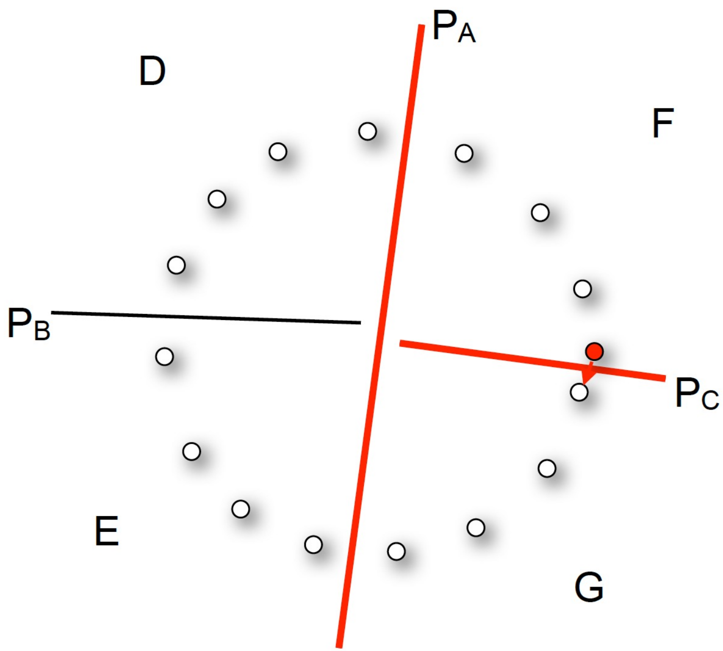


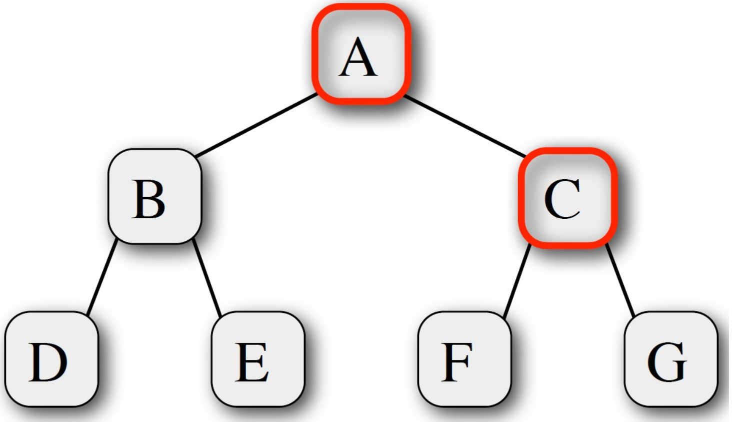
BSP Closest Point





BSP Closest Point





ICP Variants

- Variants on the following stages of ICP have been proposed:
  - Selecting sample points (from one or both meshes)
  - Matching to points in the other mesh
  - Weighting the correspondences
  - Rejecting certain (outlier) point pairs
  - Assigning an error metric to the current transform
  - Minimizing the error metric w.r.t. transformation

Real Time ICP