Iterative Closest Point (ICP)

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

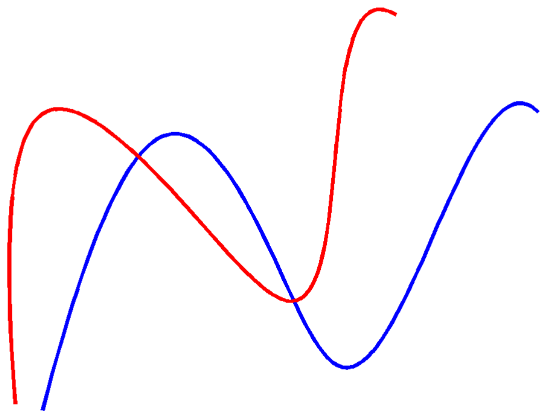
Given a pair of shapes, X and Y, iterate:

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:

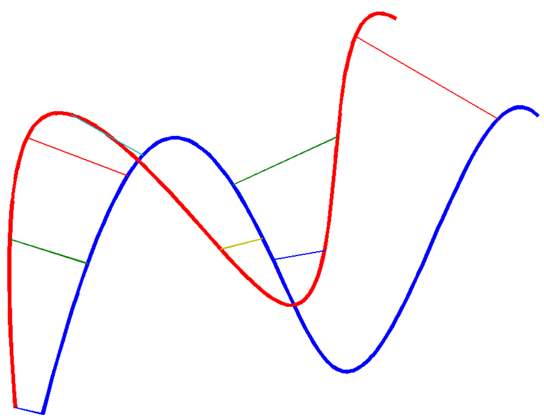


Given a pair of shapes, X and Y, iterate:

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:

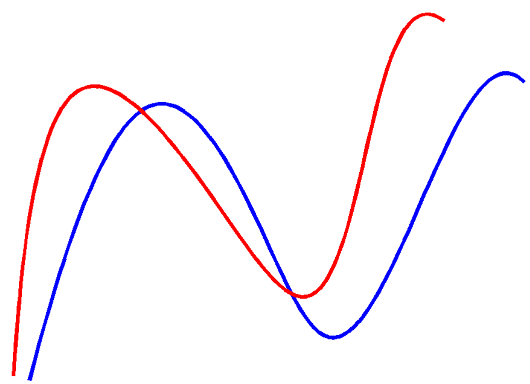


Given a pair of shapes, X and Y, iterate:

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:

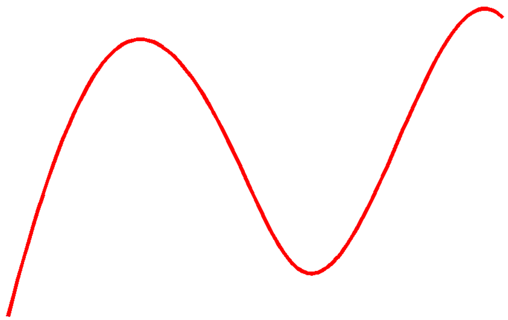


Given a pair of shapes, X and Y, iterate:

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:



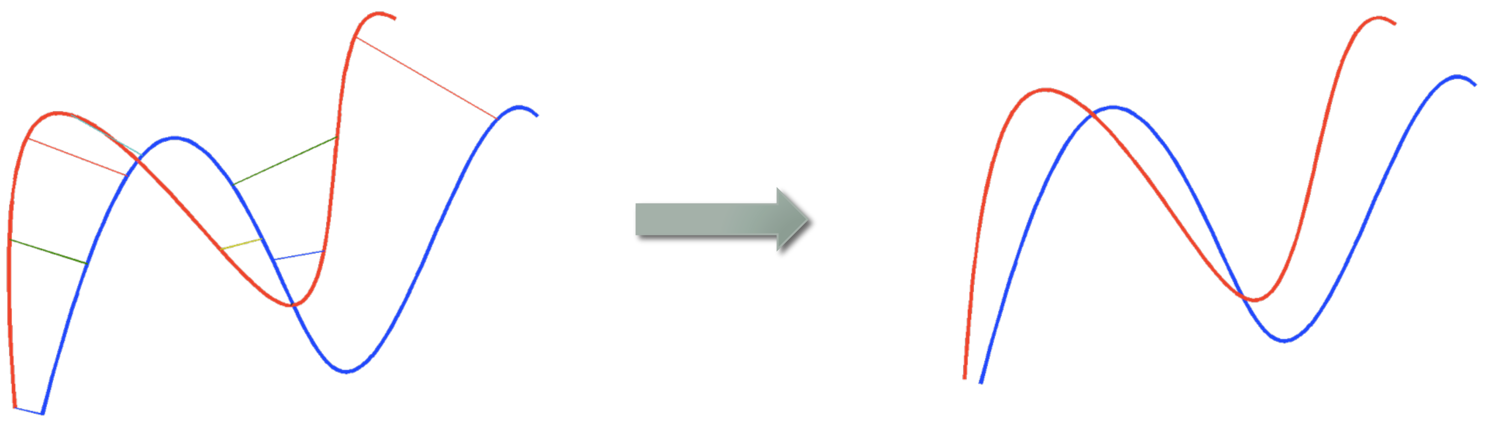
Given a pair of shapes, X and Y, iterate:

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

- 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: $\mathcal{O}(MN)$

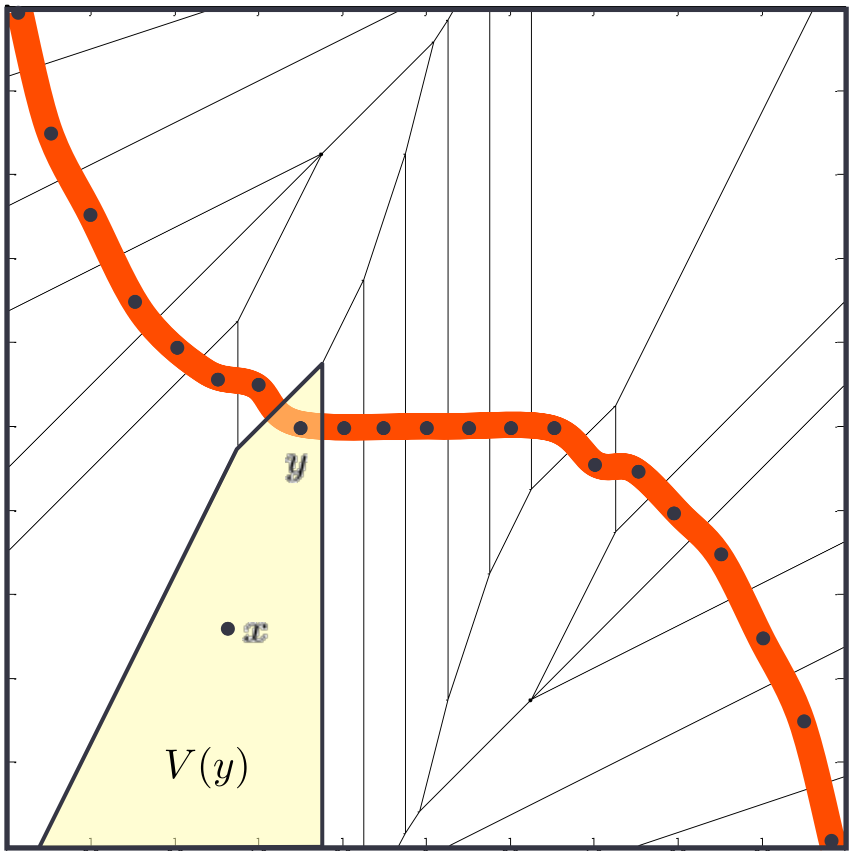
  M number of points on X, N number of points on Y.

- 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**.

Closest points: Voronoi Cells

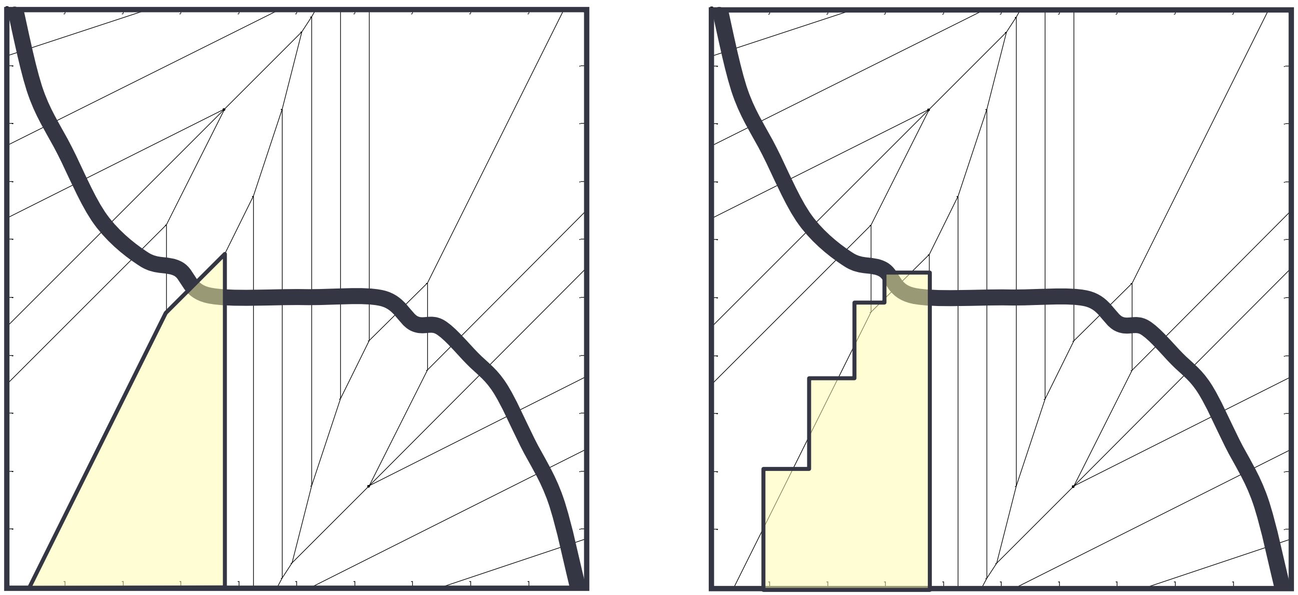


$$V(y \in Y) = \{z \in \mathbb{R}^3 : \|y - z\| < \|y' - z\| \; \forall \; y' \in Y \neq y\}$$

Source:
M. Bronstein

Closest points: Voronoi Cells

Approximate nearest neighbors



M. Bronstein

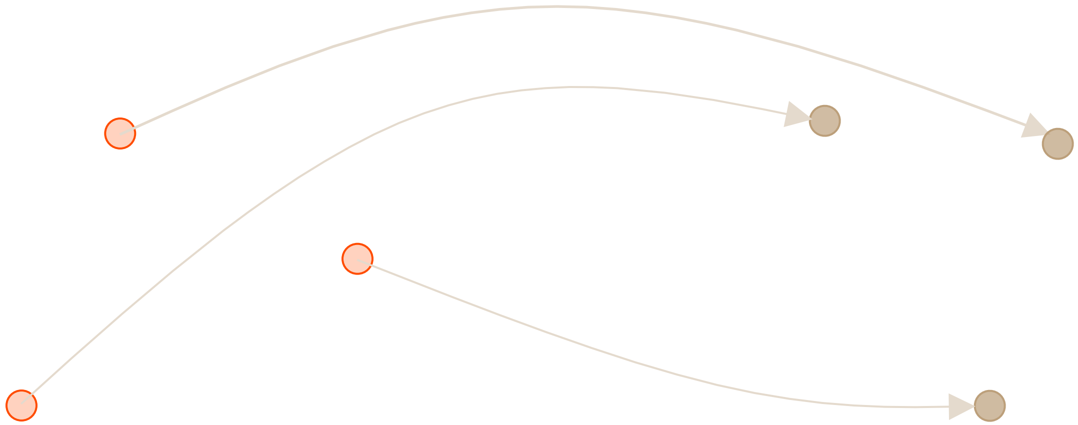
- To reduce search complexity, **approximate** Voronoi cells.
- Use **binary space partition trees** (e.g. **kd-trees** or **octrees**).
- Approximate nearest neighbor search complexity: $\mathcal{O}(N \log M)$.

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

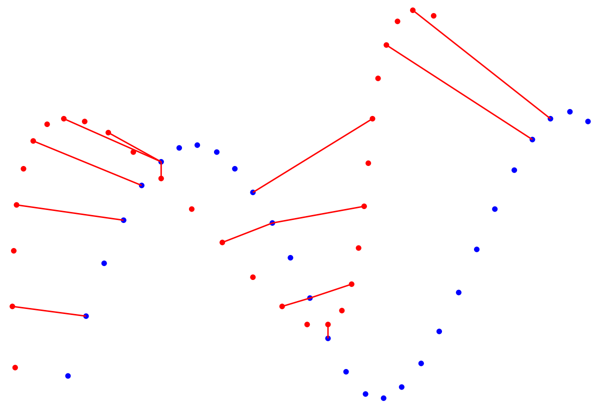
Convergence:

- 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

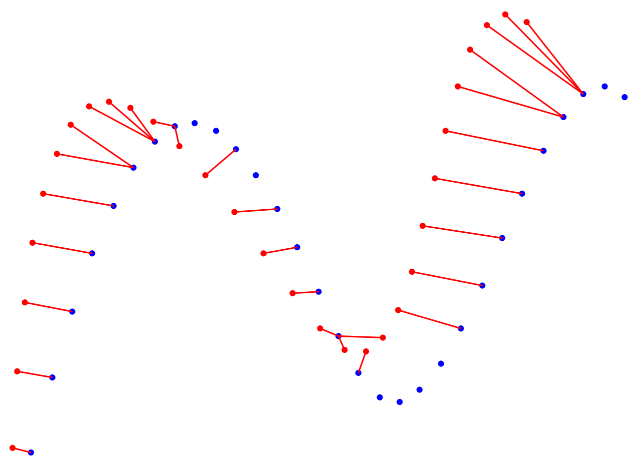


Iterative Closest Point.

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:

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



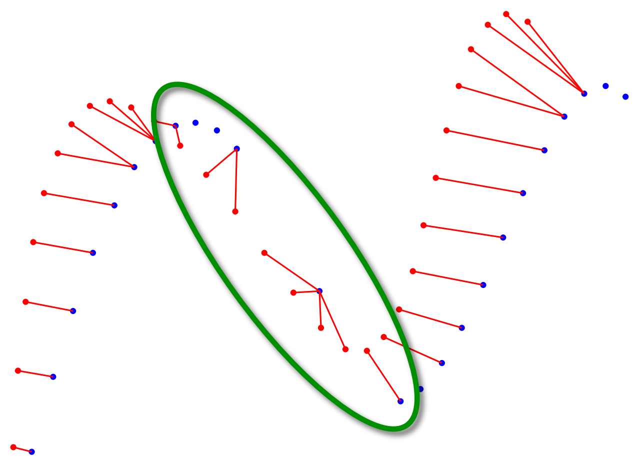
Iterative Closest Point.

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:

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

Problem:
uneven sampling



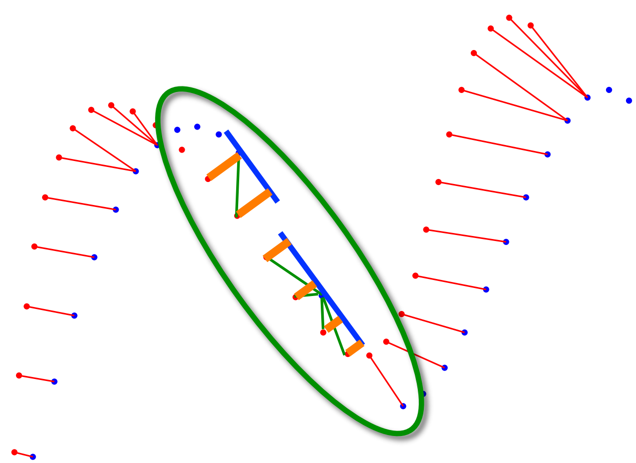
Iterative Closest Point.

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:

$$\sum_{i=1}^{N} d(\mathbf{R}x_i + t, P(y_i))^2 = \sum_{i=1}^{N} \left((\mathbf{R}x_i + t - y_i)^T \mathbf{n}_{y_i}\right)^2$$

Solution:
Minimize distance to the tangent plane



Chen, Medioni, '91

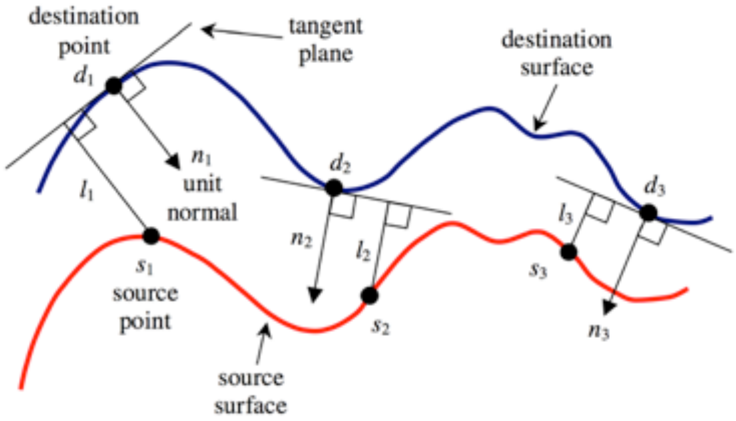
Iterative Closest Point.

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:

$$\sum_{i=1}^{N} d(\mathbf{R}x_i + t, P(y_i))^2 = \sum_{i=1}^{N} \left((\mathbf{R}x_i + t - y_i)^T \mathbf{n}_{y_i}\right)^2$$

Solution:
Minimize distance to the tangent plane



Kok-Lim Low, '04

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:

$$\mathbf{R}_{\text{opt}}, t_{\text{opt}} = \underset{\boxed{\mathbf{R}^T\mathbf{R}=\text{Id},}\ t}{\arg\min} \sum_{i=1}^{N} \left((\mathbf{R}x_i + t - y_i)^T \mathbf{n}_{y_i} \right)^2$$

Question:

How to minimize the error?

Challenge:

Although the error is **quadratic** (linear derivative), the space of rotation matrices is **not linear.**

Problem:

No closed form solution!

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:

$$\mathbf{R}_{\text{opt}}, t_{\text{opt}} = \underset{\boxed{\mathbf{R}^T\mathbf{R}=\text{Id},}\ t}{\arg\min} \sum_{i=1}^{N} \left((\mathbf{R}x_i + t - y_i)^T \mathbf{n}_{y_i} \right)^2$$

Common Solution:

Linearize rotation. Assume rotation angle is small.

$$\mathbf{R}x_i \approx x_i + r \times x_i$$

r : axis,

$\|r\|_2$: angle of rotation.

Note: follows from Rodrigues's formula

$$R(r,\alpha)x_i = x_i \cos(\alpha) + (r \times x_i)\sin(\alpha) + r(r^T x_i)(1 - \cos(\alpha))$$

And first order approximations: $\sin(\alpha) \approx \alpha, \quad \cos(\alpha) \approx 1$

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 r, t minimizing:

$$E(r,t) = \sum_{i=1}^{N} \left((x_i + r \times x_i + t - y_i)^T \mathbf{n}_{y_i}\right)^2$$
$$= \sum_{i=1}^{N} \left((x_i - y_i)^T \mathbf{n}_{y_i} + r^T (x_i \times \mathbf{n}_{y_i}) + t^T \mathbf{n}_{y_i}\right)^2$$

Setting: $\frac{\partial}{\partial r} E(r,t) = 0$ and $\frac{\partial}{\partial t} E(r,t) = 0$ leads to a 6x6 linear system

$$Ax = b$$

$$x = \begin{pmatrix} r \\ t \end{pmatrix} \quad A = \sum \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix} \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix}^T \quad b = \sum (y_i - x_i)^T \mathbf{n}_{y_i} \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix}$$