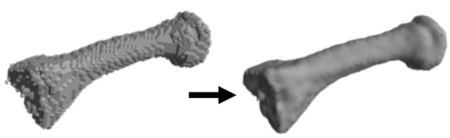
CSE 554

Lecture 7: Alignment

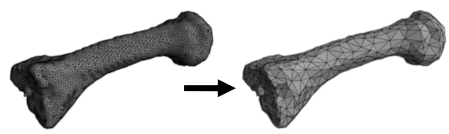
Fall 2012

Review

- Fairing (smoothing)
  - Relocating vertices to achieve a smoother appearance
  - Method: centroid averaging

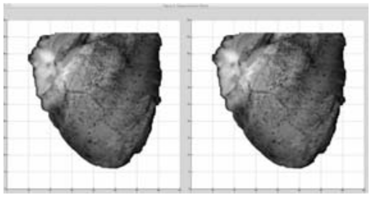


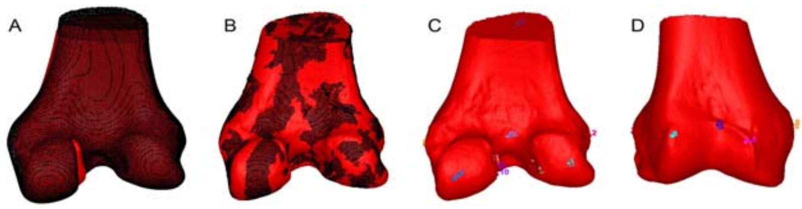
- Simplification
  - Reducing vertex count
  - Method: edge collapsing



Registration

- Fitting one model to match the shape of another
  - Automated annotation
  - Tracking and motion analysis
  - Shape and data comparison

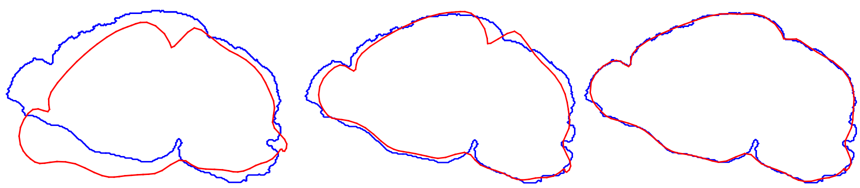




Registration

Washington University in St. Louis
School of Engineering & Applied Science

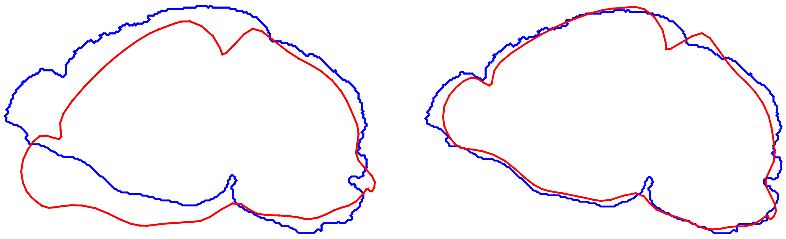
- Challenges: global and local shape differences
  - Imaging causes global shifts and tilts
    - Requires alignment
  - The shape of the organ or tissue differs in subjects and evolve over time
    - Requires deformation



Brain outlines of two mice | After alignment | After deformation

Alignment

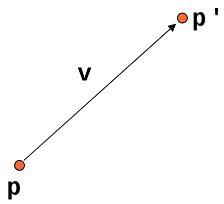
- Registration by translation or rotation
  - The structure stays "rigid" under these two transformations
    - Called rigid-body or isometric (distance-preserving) transformations
  - Mathematically, they are represented as matrix/vector operations



Before alignment　　　　After alignment

Transformation Math

- Translation
  - Vector addition: $\mathbf{p'} = \mathbf{v} + \mathbf{p}$
  - 2D: $\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}$
  - 3D: $\begin{pmatrix} p'_x \\ p'_y \\ p'_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$



Transformation Math

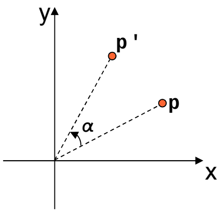
Washington University in St. Louis
School of Engineering & Applied Science

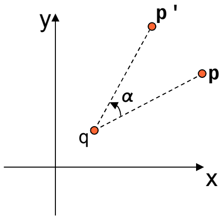
- Rotation
  - Matrix product: $\mathbf{p'} = \mathbf{R} \cdot \mathbf{p}$
  - 2D: $\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix}$

    $\mathbf{R} = \begin{pmatrix} \mathrm{Cos}[\alpha] & -\mathrm{Sin}[\alpha] \\ \mathrm{Sin}[\alpha] & \mathrm{Cos}[\alpha] \end{pmatrix}$

    - Rotate around the origin!
    - To rotate around another point q:

      $\mathbf{p'} = \mathbf{R} \cdot (\mathbf{p} - \mathbf{q}) + \mathbf{q}$





Transformation Math

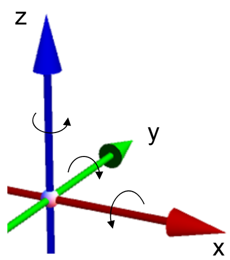
Washington University in St. Louis
School of Engineering & Applied Science

- Rotation
  - Matrix product: $\mathbf{p'} = \mathbf{R} \cdot \mathbf{p}$
  - 3D: $\begin{pmatrix} p'_x \\ p'_y \\ p'_z \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$

Around X axis: $\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{Cos}[\alpha] & -\mathrm{Sin}[\alpha] \\ 0 & \mathrm{Sin}[\alpha] & \mathrm{Cos}[\alpha] \end{pmatrix}$

Around Y axis: $\mathbf{R}_y = \begin{pmatrix} \mathrm{Cos}[a] & 0 & \mathrm{Sin}[a] \\ 0 & 1 & 0 \\ -\mathrm{Sin}[a] & 0 & \mathrm{Cos}[a] \end{pmatrix}$

Around Z axis: $\mathbf{R}_z = \begin{pmatrix} \mathrm{Cos}[a] & -\mathrm{Sin}[a] & 0 \\ \mathrm{Sin}[a] & \mathrm{Cos}[a] & 0 \\ 0 & 0 & 1 \end{pmatrix}$



Any arbitrary 3D rotation can be composed from these three rotations

Transformation Math

- Properties of an arbitrary rotational matrix
  - Orthonormal (orthogonal and normal): $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$
    - Examples:

$$\begin{pmatrix} \cos[\alpha] & -\sin[\alpha] \\ \sin[\alpha] & \cos[\alpha] \end{pmatrix} \cdot \begin{pmatrix} \cos[\alpha] & \sin[\alpha] \\ -\sin[\alpha] & \cos[\alpha] \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos[\alpha] & -\sin[\alpha] \\ 0 & \sin[\alpha] & \cos[\alpha] \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos[\alpha] & \sin[\alpha] \\ 0 & -\sin[\alpha] & \cos[\alpha] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

    - Easy to invert: $\mathbf{R}^{-1} = \mathbf{R}^T$
  - Any orthonormal matrix represents a rotation around some axis (not limited to X,Y,Z)

Transformation Math

- Properties of an arbitrary rotational matrix
  - Given an orthonormal matrix, the angle of rotation represented by the matrix can be easily calculated from the trace of the matrix
    - Trace: sum of diagonal entries
    - 2D: The trace equals 2 Cos(a), where a is the rotation angle
    - 3D: The trace equals 1 + 2 Cos(a)
  - The larger the trace, the smaller the rotation angle

Transformation Math

Washington University in St.Louis
School of Engineering & Applied Science

- Eigenvectors and eigenvalues
  - Let M be a square matrix, v is an eigenvector and λ is an eigenvalue if:

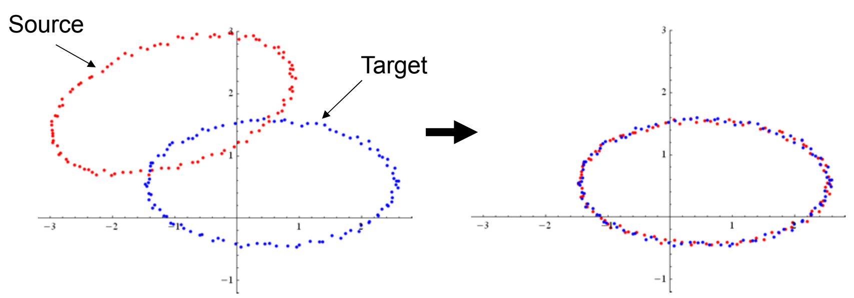
$$\mathbf{M} \cdot \mathbf{v} = \lambda * \mathbf{v}$$

    - If M represents a rotation (i.e., orthonormal), the rotation axis is an eigenvector whose eigenvalue is 1.
  - There are at most m distinct eigenvalues for a m by m matrix
  - Any scalar multiples of an eigenvector is also an eigenvector (with the same eigenvalue).

Alignment

Washington University in St.Louis
School of Engineering & Applied Science

- Input: two models represented as point sets
  - Source and target
- Output: locations of the translated and rotated source points



Alignment

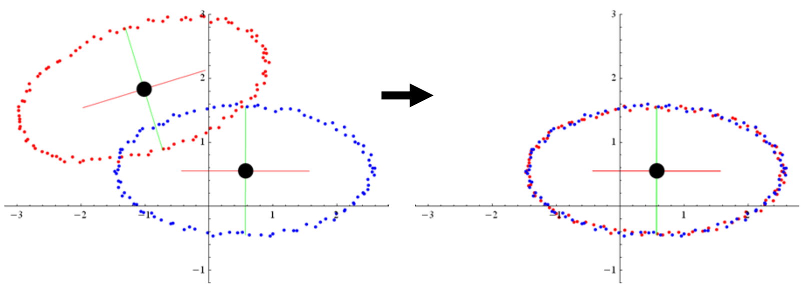
Washington University in St.Louis
School of Engineering & Applied Science

- Method 1: Principal component analysis (PCA)
  - Aligning principal directions
- Method 2: Singular value decomposition (SVD)
  - Optimal alignment given prior knowledge of correspondence
- Method 3: Iterative closest point (ICP)
  - An iterative SVD algorithm that computes correspondences as it goes

Method 1: PCA

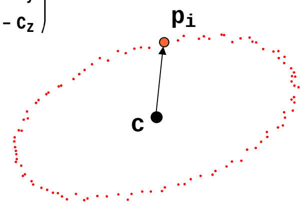
Washington University in St.Louis
School of Engineering & Applied Science

- Compute a shape-aware coordinate system for each model
  - Origin: Centroid of all points
  - Axes: Directions in which the model varies most or least
- Transform the source to align its origin/axes with the target



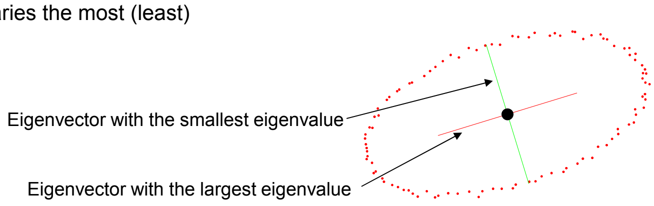
Method 1: PCA

- Computing axes: Principal Component Analysis (PCA)
  - Consider a set of points $p_1, \ldots, p_n$ with centroid location c
    - Construct matrix P whose *i*-th column is vector $p_i - c$
      - 2D (2 by n): $P = \begin{pmatrix} p_{1_x} - c_x & p_{2_x} - c_x & \cdots & p_{n_x} - c_x \\ p_{1_y} - c_y & p_{2_y} - c_y & \cdots & p_{n_y} - c_y \end{pmatrix}$
      - 3D (3 by n): $P = \begin{pmatrix} p_{1_x} - c_x & p_{2_x} - c_x & \cdots & p_{n_x} - c_x \\ p_{1_y} - c_y & p_{2_y} - c_y & \cdots & p_{n_y} - c_y \\ p_{1_z} - c_z & p_{2_z} - c_z & \cdots & p_{n_z} - c_z \end{pmatrix}$
    - Build the covariance matrix: $M = P \cdot P^T$
      - 2D: a 2 by 2 matrix
      - 3D: a 3 by 3 matrix



Method 1: PCA

- Computing axes: Principal Component Analysis (PCA)
  - Eigenvectors of the covariance matrix represent principal directions of shape variation
    - The eigenvectors are un-singed and orthogonal (2 in 2D; 3 in 3D)
  - Eigenvalues indicate amount of variation along each eigenvector
    - Eigenvector with largest (smallest) eigenvalue is the direction where the model shape varies the most (least)



Method 1: PCA

- PCA-based alignment
  - Let c_S, c_T be centroids of source and target.
  - First, translate source to align c_S with c_T:

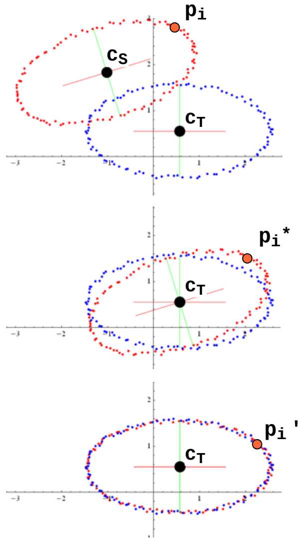
    $$p_i^* = p_i + (c_T - c_S)$$

  - Next, find rotation R that aligns two sets of PCA axes, and rotate source around c_T:

    $$p_i' = c_T + R \cdot (p_i^* - c_T)$$

  - Combined:

    $$p_i' = c_T + R \cdot (p_i - c_S)$$



Method 1: PCA

- Finding rotation between two sets of oriented axes
  - Let A, B be two matrices whose columns are the axes
    - The axes are orthogonal and normalized (i.e., both A and B are orthonormal)
  - We wish to compute a rotation matrix R such that:

    $$R \cdot A = B$$

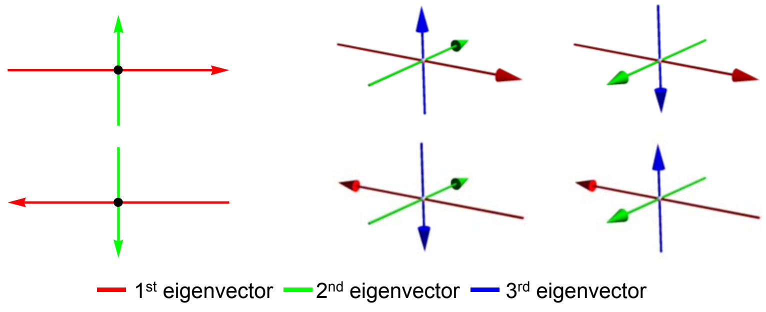
  - Notice that A and B are orthonormal, so we have:

    $$R = B \cdot A^{-1} = B \cdot A^T$$

Method 1: PCA

Washington University in St. Louis
School of Engineering & Applied Science

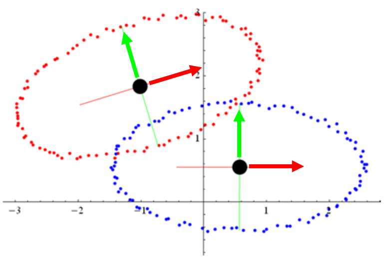
- Assigning orientation to PCA axes
  - There are 2 possible orientation assignments in 2D
  - In 3D, there are 4 possibilities (observing the right-hand rule)



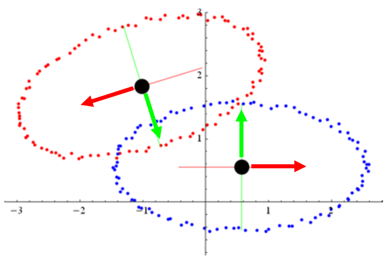
Method 1: PCA

Washington University in St. Louis
School of Engineering & Applied Science

- Finding rotation between two sets of un-oriented axes
  - Fix the orientation of the target axes.
  - For each orientation assignment of the source axes, compute R
  - Pick the R with smallest rotation angle (by checking the trace of R)



Smaller rotation

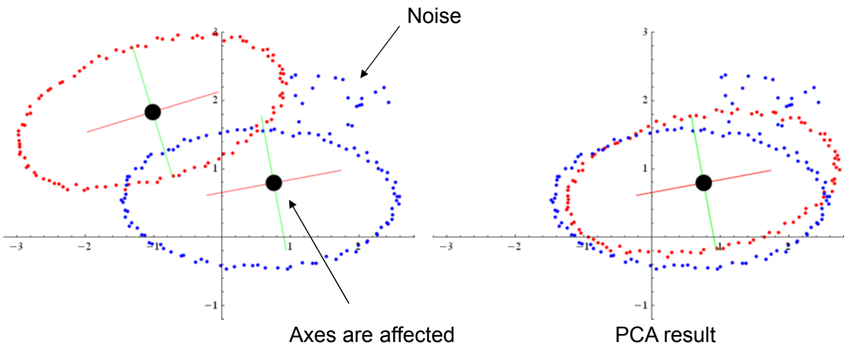


Larger rotation

Method 1: PCA

Washington University in St. Louis
School of Engineering & Applied Science

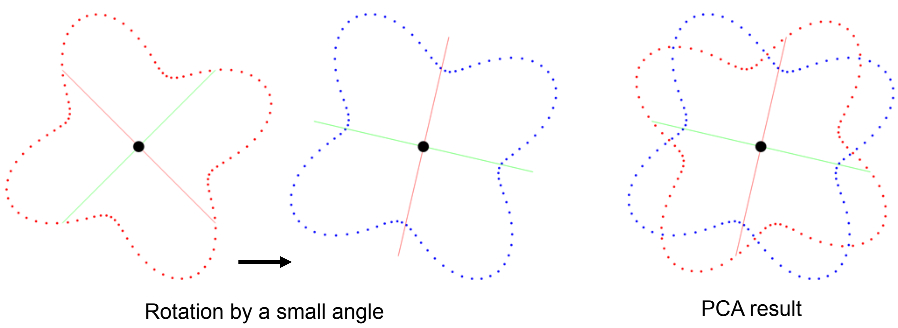
- Limitations
  - Centroid and axes are affected by noise



Method 1: PCA

Washington University in St. Louis
School of Engineering & Applied Science

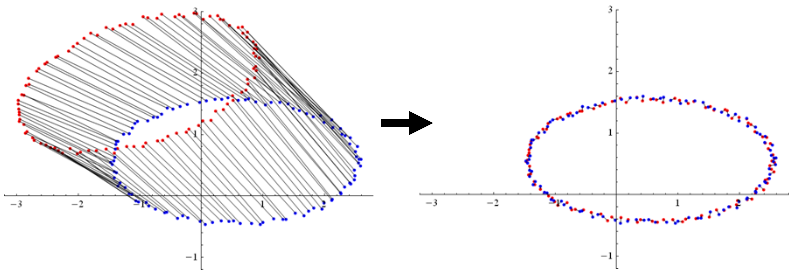
- Limitations
  - Axes can be unreliable for circular objects
    - Eigenvalues become similar, and eigenvectors become unstable



Method 2: SVD

Washington University in St.Louis
School of Engineering & Applied Science

- Optimal alignment between corresponding points
  - Assuming that for each source point, we know where the corresponding target point is



Method 2: SVD

Washington University in St.Louis
School of Engineering & Applied Science

- Formulating the problem
  - Source points $p_1,\ldots,p_n$ with centroid location c_S
  - Target points $q_1,\ldots,q_n$ with centroid location c_T
    - q_i is the corresponding point of p_i
  - After centroid alignment and rotation by some R, a transformed source point is located at:

$$p_i' = c_T + R \cdot (p_i - c_S)$$

  - We wish to find the R that minimizes sum of pair-wise distances:

$$E = \sum_{i=1}^{n} \| q_i - p_i' \|^2$$

Method 2: SVD

Washington University in St. Louis
School of Engineering & Applied Science

- An equivalent formulation
  - Let P be a matrix whose *i*-th column is vector $p_i - c_S$
  - Let Q be a matrix whose *i*-th column is vector $q_i - c_T$
  - Consider the cross-covariance matrix:

$$\mathbf{M} = \mathbf{P} \cdot \mathbf{Q}^T$$

  - Find the orthonormal matrix R that maximizes the trace:

$$\mathrm{Tr}[\mathbf{R} \cdot \mathbf{M}]$$

Method 2: SVD

Washington University in St. Louis
School of Engineering & Applied Science

- Solving the minimization problem
  - Singular value decomposition (SVD) of an m by m matrix M:

$$\mathbf{M} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T$$

    - U,V are m by m orthonormal matrices (i.e., rotations)
    - W is a diagonal m by m matrix with non-negative entries
  - The orthonormal matrix (rotation) $\mathbf{R} = \mathbf{V} \cdot \mathbf{U}^T$ is the R that maximizes the trace $\mathrm{Tr}[\mathbf{R} \cdot \mathbf{M}]$
  - SVD is available in *Mathematica* and many Java/C++ libraries

Method 2: SVD

- SVD-based alignment: summary
  - Forming the cross-covariance matrix

    $$\mathbf{M} = \mathbf{P} \cdot \mathbf{Q}^T$$

  - Computing SVD

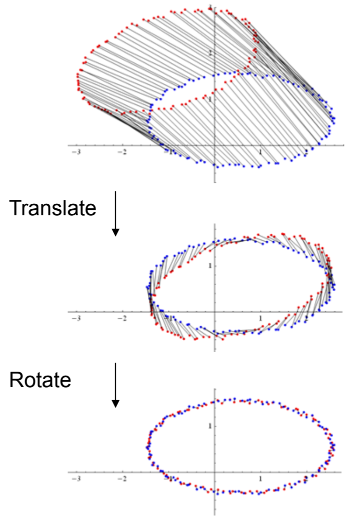
    $$\mathbf{M} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T$$

  - The rotation matrix is

    $$\mathbf{R} = \mathbf{V} \cdot \mathbf{U}^T$$

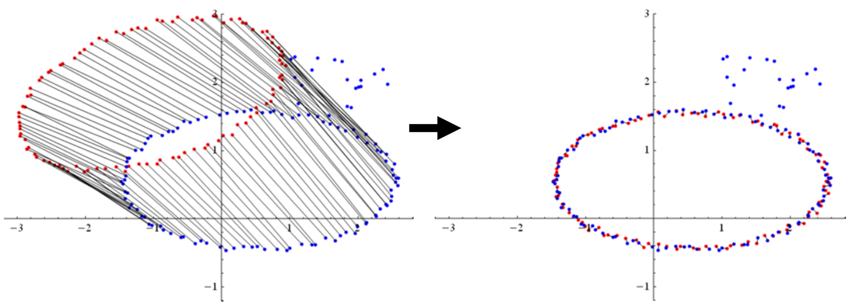
  - Translate and rotate the source:

    $$\mathbf{p_i}' = \mathbf{c_T} + \mathbf{R} \cdot (\mathbf{p_i} - \mathbf{c_S})$$



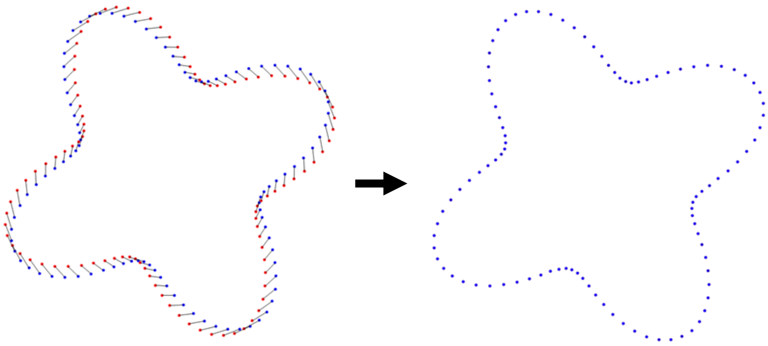
Method 2: SVD

- Advantage over PCA: more stable
  - As long as the correspondences are correct



Method 2: SVD

- Advantage over PCA: more stable
  - As long as the correspondences are correct



CSE554 Alignment Slide 29

Method 2: SVD

- Limitation: requires accurate correspondences
  - Which are usually not available

CSE554 Alignment Slide 30

Method 3: ICP

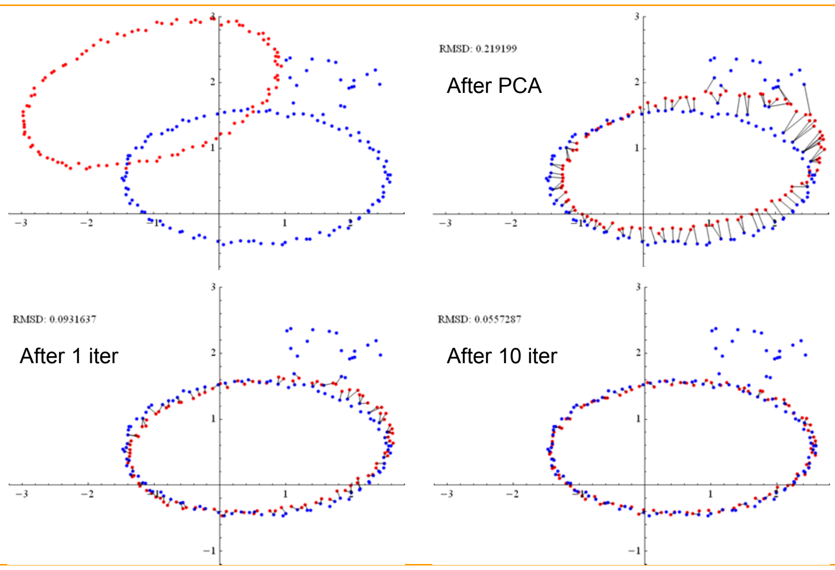
Washington University in St.Louis
School of Engineering & Applied Science

- The idea
  - Use PCA alignment to obtain initial guess of correspondences
  - Iteratively improve the correspondences after repeated SVD
- Iterative closest point (ICP)
  - 1. Transform the source by PCA-based alignment
  - 2. For each transformed source point, assign the closest target point as its corresponding point. Align source and target by SVD.
    - Not all target points need to be used
  - 3. Repeat step (2) until a termination criteria is met.

CSE554 Alignment Slide 31

ICP Algorithm

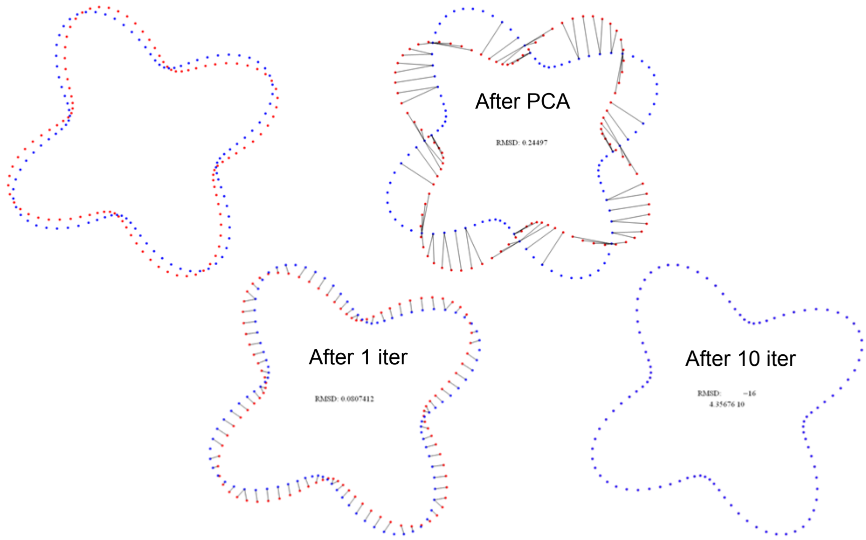
Washington University in St.Louis
School of Engineering & Applied Science



CSE554 Alignment Slide 32

ICP Algorithm

Washington University in St.Louis
School of Engineering & Applied Science



ICP Algorithm

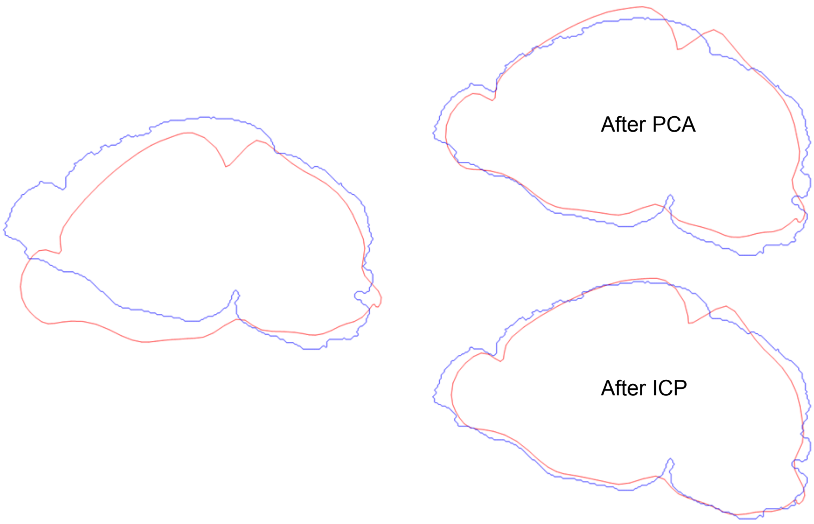
Washington University in St.Louis
School of Engineering & Applied Science

- Termination criteria
  - A user-given maximum iteration is reached
  - The improvement of fitting is small
    - Root Mean Squared Distance (RMSD):

$$\sqrt{\frac{\sum_{i=1}^{n} \|q_i - p_i'\|^2}{n}}$$

      - Captures average deviation in all corresponding pairs
    - Stops the iteration if the difference in RMSD before and after each iteration falls beneath a user-given threshold

More Examples



More Examples

Washington University in St. Louis
School of Engineering & Applied Science