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

Brain outlines of two mice
A  B  C  D
After alignment
After deformation

Registration

• 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
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

- Rotation
  - Matrix product: \( \mathbf{p}' = \mathbf{R} \cdot \mathbf{p} \)
  
  - 2D:
    \[
    \begin{pmatrix}
    \mathbf{p}'_x \\
    \mathbf{p}'_y
    \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix}
    \mathbf{p}_x \\
    \mathbf{p}_y
    \end{pmatrix}
    \]
    
    \[
    \mathbf{R} = \begin{pmatrix}
    \cos(\alpha) & -\sin(\alpha) \\
    \sin(\alpha) & \cos(\alpha)
    \end{pmatrix}
    \]
    
    - Rotate around the origin!
    - To rotate around another point \( \mathbf{q} \):
      
      \[
      \mathbf{p}' = \mathbf{R} \cdot (\mathbf{p} - \mathbf{q}) + \mathbf{q}
      \]

- 3D:
  
  \[
  \begin{pmatrix}
  \mathbf{p}'_x \\
  \mathbf{p}'_y \\
  \mathbf{p}'_z
  \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix}
  \mathbf{p}_x \\
  \mathbf{p}_y \\
  \mathbf{p}_z
  \end{pmatrix}
  \]

  Around X axis: \( \mathbf{R}_x = \begin{pmatrix}
  1 & 0 & 0 \\
  0 & \cos(\alpha) & -\sin(\alpha) \\
  0 & \sin(\alpha) & \cos(\alpha)
  \end{pmatrix} \)

  Around Y axis: \( \mathbf{R}_y = \begin{pmatrix}
  \cos(\alpha) & 0 & \sin(\alpha) \\
  0 & 1 & 0 \\
  -\sin(\alpha) & 0 & \cos(\alpha)
  \end{pmatrix} \)

  Around Z axis: \( \mathbf{R}_z = \begin{pmatrix}
  \cos(\alpha) & -\sin(\alpha) & 0 \\
  \sin(\alpha) & \cos(\alpha) & 0 \\
  0 & 0 & 1
  \end{pmatrix} \)

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

Properties of an arbitrary rotational matrix

- **Orthonormal** (orthogonal and normal): \( R \cdot R^T = I \)
  - Examples:
    \[
    \begin{pmatrix}
    \cos(\alpha) & -\sin(\alpha) \\
    \sin(\alpha) & \cos(\alpha)
    \end{pmatrix}
    \begin{pmatrix}
    \cos(\alpha) & \sin(\alpha) \\
    -\sin(\alpha) & \cos(\alpha)
    \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
    \]
    \[
    \begin{pmatrix}
    1 & 0 & 0 \\
    0 & \cos(\alpha) & -\sin(\alpha) \\
    0 & \sin(\alpha) & \cos(\alpha)
    \end{pmatrix}
    \begin{pmatrix}
    1 & 0 & 0 \\
    0 & \cos(\alpha) & \sin(\alpha) \\
    0 & -\sin(\alpha) & \cos(\alpha)
    \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
    \]
  - **Easy to invert**: \( R^{-1} = R^T \)
  - Any orthonormal matrix represents a rotation around some axis (not limited to X,Y,Z)

- **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

• Eigenvectors and eigenvalues
  – Let $M$ be a square matrix, $v$ is an eigenvector and $\lambda$ is an eigenvalue if:
    \[ M \cdot v = \lambda \cdot v \]
  – 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 $m \times m$ matrix
  – Any scalar multiples of an eigenvector is also an eigenvector (with the same eigenvalue).

Alignment

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

- **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**

- 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 \( n \)): \( P = \begin{bmatrix} p_{1x} - c_x & p_{2x} - c_x & \cdots & p_{nx} - c_x \\ p_{1y} - c_y & p_{2y} - c_y & \cdots & p_{ny} - c_y \end{bmatrix} \)
      - 3D (3 by \( n \)): \( P = \begin{bmatrix} p_{1x} - c_x & p_{2x} - c_x & \cdots & p_{nx} - c_x \\ p_{1y} - c_y & p_{2y} - c_y & \cdots & p_{ny} - c_y \\ p_{1z} - c_z & p_{2z} - c_z & \cdots & p_{nz} - c_z \end{bmatrix} \)
    - 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

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

- 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)

![Diagram showing orientation and rotation](image-url)
Method 1: PCA

- Limitations
  - Centroid and axes are affected by noise

![PCA result with noise](image1)

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

![Rotation by a small angle](image2)
Method 2: SVD

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

- 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

- 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:
    $$M = P \cdot Q^T$$
  - Find the orthonormal matrix $R$ that maximizes the trace:
    $$\text{Tr} [R \cdot M]$$

Method 2: SVD

- Solving the minimization problem
  - Singular value decomposition (SVD) of an $m$ by $m$ matrix $M$:
    $$M = U \cdot W \cdot V^T$$
    - $U, V$ 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) $R = V \cdot U^T$ is the $R$ that maximizes the trace $\text{Tr} [R \cdot M]$
  - SVD is available in Mathematica and many Java/C++ libraries
Method 2: SVD

- SVD-based alignment: summary
  - Forming the cross-covariance matrix
    \[ M = P \cdot Q^T \]
  - Computing SVD
    \[ M = U \cdot W \cdot V^T \]
  - The rotation matrix is
    \[ R = V \cdot U^T \]
  - Translate and rotate the source:
    \[ p_i' = c_T + R \cdot (p_i - c_S) \]

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

Method 2: SVD

- Limitation: requires accurate correspondences
  - Which are usually not available
Method 3: ICP

- 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.

ICP Algorithm

After PCA

After 1 iter

After 10 iter
ICP Algorithm

- 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

After PCA

After ICP

More Examples

After PCA

After ICP