3D Scanning & Motion Capture

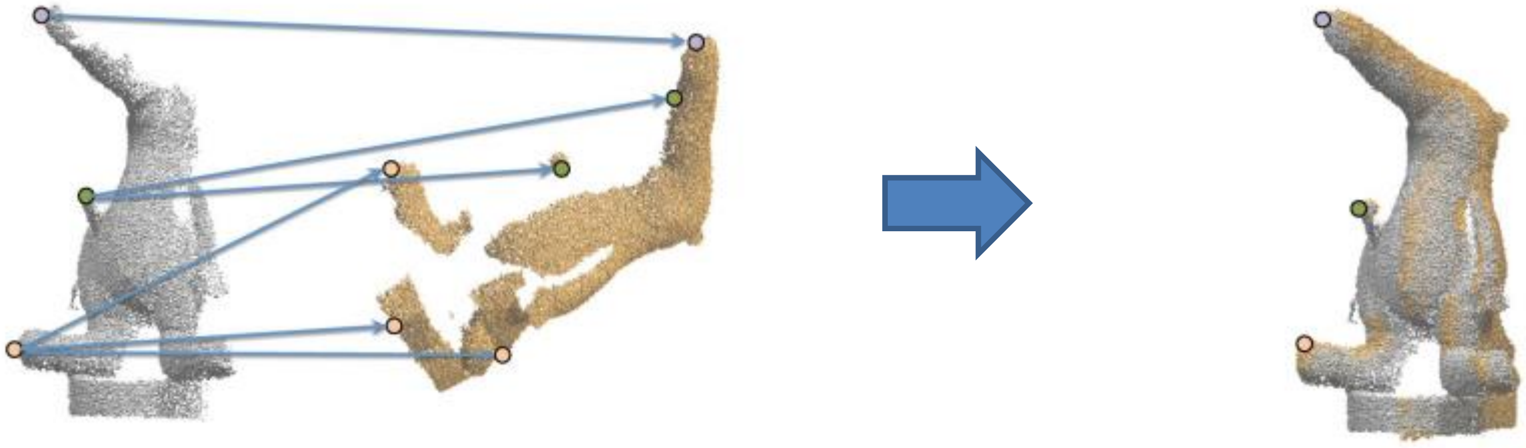
Exercise - 3

Armen Avetisyan, Dejan Azinović

Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

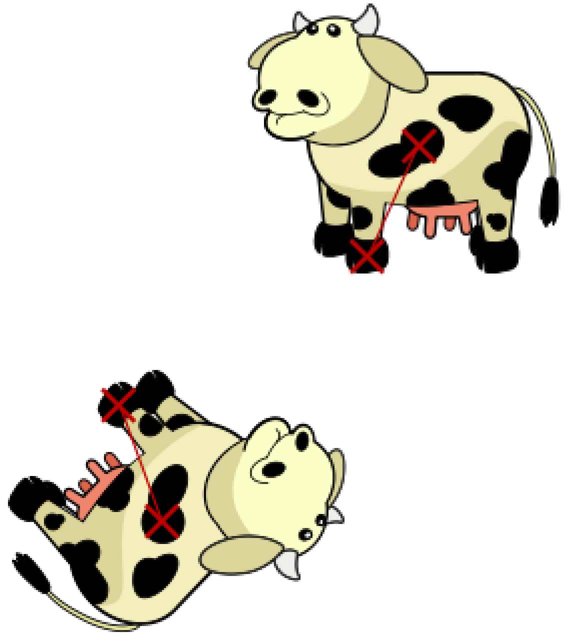


Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

  - Compute center of gravity of both objects
  - Scale one object to match the avg. distance from all vertices to the center of gravity



Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

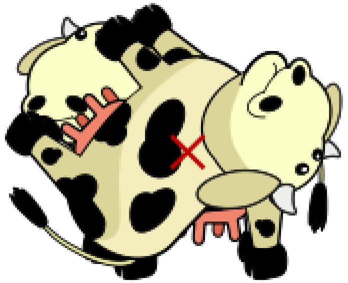
  - Translation is given by the vector between the center of gravity of both objects

Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

  - Assume objects that are zero centered
    - Target object: $\{x_0, .. x_{n-1}\}$
    - Moving object: $\{\hat{x}_0\ , .. \hat{x}_{n-1}\}$



$$\sum_i \|x_i - R \cdot \hat{x}_i\|_2^2 \ \rightarrow min$$

$$\left\|X - \hat{X}R^T\right\|_F^2 \rightarrow min$$

Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

$$\left\| X - \hat{X}R^T \right\|_F^2 \to min$$

$\|A\|_F^2 = trace(A^T A)$

Cyclic invariance of trace:

$trace(ABC) = trace(CAB)$

$$\left\| X - \hat{X}R^T \right\|_F^2 = trace\left(X^T X - X^T \hat{X} R^T - \left(\hat{X}R^T\right)^T X + \left(\hat{X}R^T\right)^T \left(\hat{X}R^T\right)\right) \to min$$

$$trace\left(-X^T \hat{X} R^T - \left(\hat{X}R^T\right)^T X + \left(\hat{X}R^T\right)^T \left(\hat{X}R^T\right)\right) \to min$$

$$-2 \cdot trace\left(X^T \hat{X} R^T\right) \to min$$

$$trace\left(X^T \hat{X} R^T\right) \to max$$

$SVD\colon X^T \hat{X} = USV^T$

$$trace\left(USV^T R^T\right) \to max$$

$$trace\left(S \underbrace{V^T R^T U}\right) \to max$$

Singular values
→positive

Product of orthogonal matrices
→max if equal to Identity

Procrustes

- Problem: Align two objects using known correspondences

  →scaling, translation, rotation

$$\left\| X - \hat{X} R^T \right\|_F^2 \rightarrow min$$

  - Compute SVD of the Cross-Covariance Matrix

$$X^T \hat{X} = USV^T$$

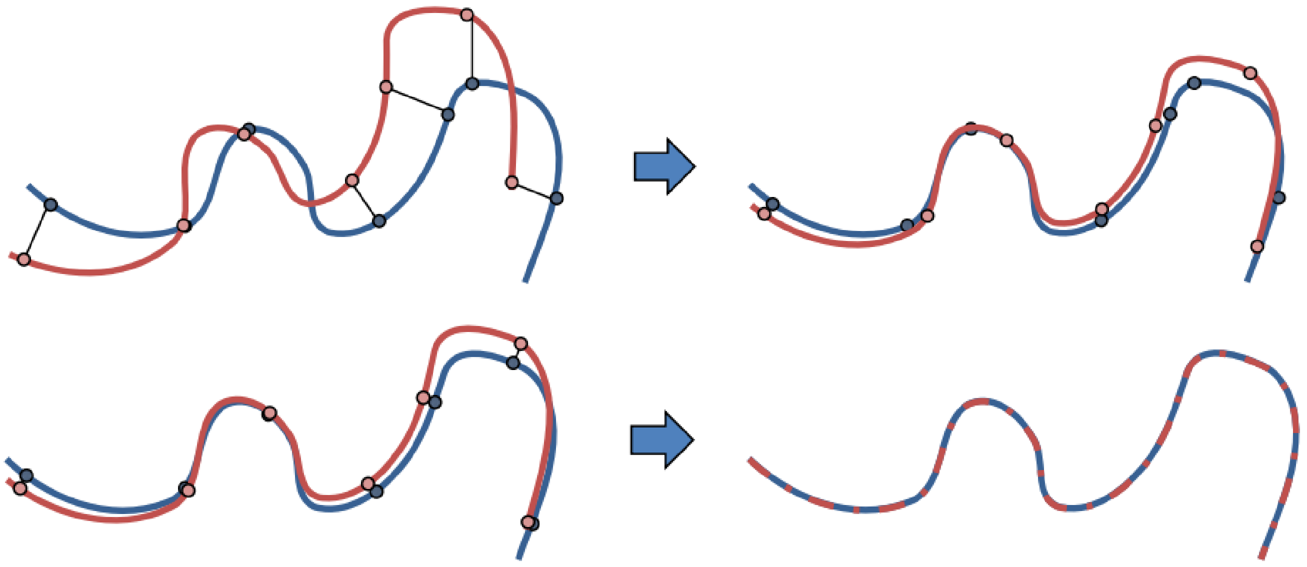
  - Compute the rotation

$$R = UV^T$$

ICP (Iterative Closest Point)

- Problem: Align two objects with unknown correspondences
  - Iterate:
    - Estimate correspondences using the current alignment and nearest neighbors
    - Use the correspondences to compute new alignment based on
      - Point-to-point distances
        - » Procrustes

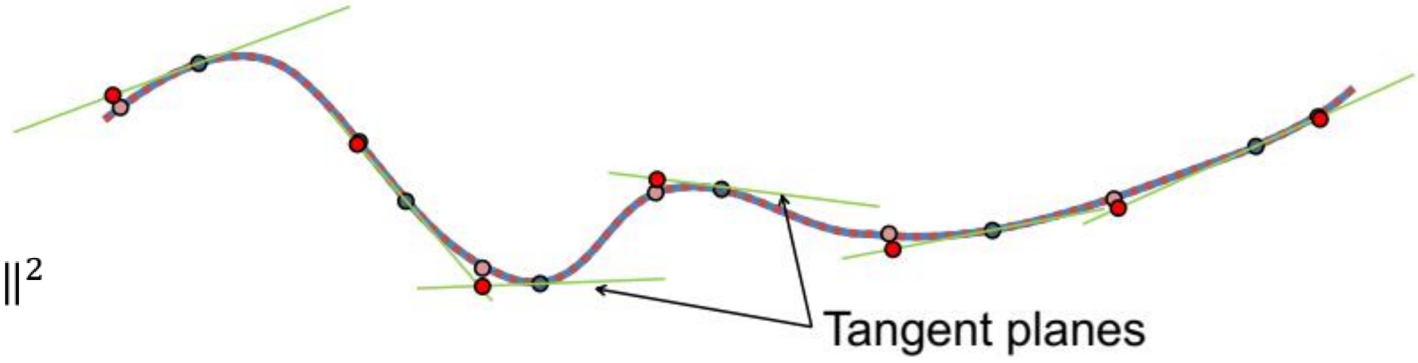
$$\min_{R,t} \sum_i \|p_i - (Rq_i + t)\|^2$$



ICP (Iterative Closest Point)

- Problem: Align two objects with unknown correspondences
  - Iterate:
    - Estimate correspondences using the current alignment and nearest neighbors
    - Use the correspondences to compute new alignment based on
      - Point-to-point distances
      - Point-to-plane distances
        - » Faster convergence
        - » Non-linear!

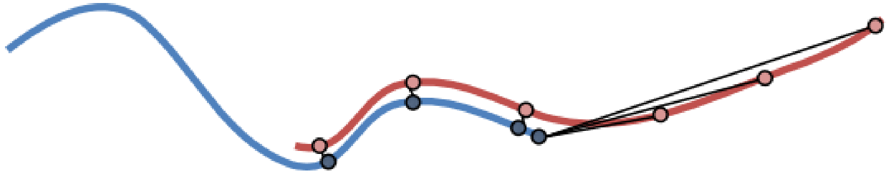
$$\min_{R,t} \sum_i \| (p_i - (Rq_i + t)) \cdot n_i \|^2$$



ICP (Iterative Closest Point)

- Problem: Align two objects with unknown correspondences
  - Iterate:
    - Estimate correspondences using the current alignment and nearest neighbors
    - Use the correspondences to compute new alignment based on
      - Point-to-point distances
      - Point-to-plane distances
    - Use weighting of correspondences and pruning
      - Good correspondences are close, have similar normal, …
      - Prune correspondences to border

$$\min_{R,t} \sum_i w_i \|p_i - (Rq_i + t)\|^2$$



ICP (Iterative Closest Point)

- Point-to-plane distances make the problem non-linear. It is possible to linearize it:
  https://www-new.comp.nus.edu.sg/~lowkl/publications/lowk_point-toplane_icp_techrep.pdf
- Both point-to-plane and point-to-point constraints can be included in a single system matrix. Point-to-point constraints must satisfy the following equation:

$$\begin{bmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x \\ s_y \\ s_z \\ 1 \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z \\ 1 \end{bmatrix}$$

- where (alpha, beta, gamma) represents the rotation angles, *t* is the translation vector, *s* is the source point and *d* the destination point. Three constraints are to be added for every point (one for every coordinate of the point).