3D Shape Analysis

190070033, 190070068, 19D070003

April 2021

(a) Procrustes distance

The Procrustes distance between two pointset-represented shapes $\mathbf{z_1}$ and $\mathbf{z_2}$, which are each N vectors having 3 co-ordinates each, is defined as-

$$\min_{R,T,s} \sum_{n=1}^{N} ||z_{1n} - sRz_{2n} - T||^2$$

T is a Translation vector having 3 elements.
R is a 3×3 orthonormal (Rotation) matrix. This places 3 Normalisation conditions ($||r_i|| = 1$), and 3 pairwise orthogonality ($< r_i, r_j >= 0$) conditions on its elements.
s is a size factor (scalar).

Distance calculation

1. Translate both shapes such that their centroids lie on the origin. This effectively removes the T term, and results in the distance obtained with the optimized T. Let these be z_1', z_2'
2. Fix size factor s, optimize rotation using Kabsch Algorithm. Let the points of the translated and scaled pointsets (i.e. sz_2', z_1')be X, Y, which are $3 \times N$ matrices. $\mathbf{x_i}$, the i^{th} point of X will be a row vector with 3 elements. We intend to minimize

$$\begin{aligned}
&= argmin_R \sum_{i=1}^{N} ||R\mathbf{x_i} - \mathbf{y_i}||^2 \\
&= argmin_R \sum_{i=1}^{N} (\mathbf{x_i^T x_i} - 2\mathbf{y_i^T} R\mathbf{x_i} + \mathbf{y_i^T y_i} \\
&R^T R = I \text{ Orthonormal matrix} \\
&= argmax_R \sum_{i=1}^{N} \mathbf{y_i^T} R\mathbf{x_i} \\
&= argmax_R tr(Y^T RX) \\
&= argmax_R tr(RXY^T)
\end{aligned}$$

We obtain the singular value decomposition of XY^T as $U\Sigma V^T$. Then, by the Kabsch algorithm, we set R as VU^T if $det(VU^T) = 1$ and VAU^T otherwise, where $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

3. Fixing the rotation matrix R, we optimize s. Let the points of the translated and rotated pointsets (i.e. Rz_2', z_1')be X, Y, which are $3 \times N$ matrices. $\mathbf{x_i}$, the i^{th} point of X will be a row vector with 3 elements.

$$\begin{aligned}
&= argmin_s \sum_{i=1}^{N} ||s\mathbf{x_i} - \mathbf{y_i}||^2 \\
&= argmin_s \sum_{i=1}^{N} (s^2 \mathbf{x_i^T x_i} - 2s\mathbf{y_i^T x_i} + \mathbf{y_i^T y_i} \\
&= argmin_s \; s^2 \sum_{i=1}^{N} (\mathbf{x_i^T x_i}) - 2s \sum_{i=1}^{N} (\mathbf{y_i^T x_i}) \\
&= argmin_s \; as^2 - 2bs
\end{aligned}$$

We set the value for s as b/a where $a = \sum_{i=1}^{N} \mathbf{x_i^T x_i}$ and $b = \sum_{i=1}^{N} \mathbf{y_i^T x_i}$

4. We repeat steps 2 and 3 iteratively until the change(difference after update) in both of them goes below a set threshold.

(b) Objective Function

We need to design an objective function that is less sensitive to outliers. We achieve this by setting a threshold (a hyperparameter, β) on the Procrustes distance of the points from the cluster centroid to be included in the objective function. Thus, our objective function for the cost of a clustering is-

$$O = \sum_{i=1}^{K} \frac{1}{n_i - o_i} \sum_{j} d(\mathbf{z_{ij}})$$

where

$$d(\mathbf{z_{ij}}) = \begin{cases} \text{Procrustes distance}^2 & (\mu_{\mathbf{i}}, \mathbf{z_j}) \text{ if distance } < \beta \text{ (median Procrustes}(z_j, \mu_i)) \\ 0 & \text{otherwise} \end{cases}$$

n_i is the number of "inliers" in a cluster i, while o_i is the number of outliers.

(c) Algorithm

Initialisation

Due to outliers, we do not use farthest points as the initial cluster centroids. We use KMeans++ to initliaze the cluster centroids.

1. Choose $\mathbf{z}$ uniformly at random from the shapes.
2. From the remaining shapes in the set, choose one randomly with a probability proportional to the square of the minimum distance from the chosen centroids.

We choose 3 such shapes to be the initial cluster centroids.

Update Algorithm

1. For all the shapes in the space, assign a cluster to by calculating the Procrustes distance from the 3 centroids, and choosing the cluster with the minimum value.
2. We update the centroids by calculating the means of the "inliers" of each cluster (i.e. the shapes inside the β threshold. For this, we have an algorithm.

Mean Calculation

Take a cluster with the centroid as z_n and M inliers.

$$\min z_n \sum_{m=1}^{M} \min_{R_m, T_m, s_m} \sum_{n=1}^{N} ||z_n - s_m R_m z_{mn} - T_m||^2$$

1. Translate all the shapes in the inliers to get the centroids at the origin. This effectively removes the T_m term from the optimization.
2. Fix z, and optimize R_m, T_m, s_m independently using the distance calculation algorithm for Procrustes distance between the shapes z and z_m.
3. Fix R_m, T_m, s_m, and optimize z by setting it to

$$z = \frac{\sum_{m=1}^{M}(s_m R_m z_m + T_m)}{|| \sum_{m=1}^{M}(s_m R_m z_m + T_m)||}$$

Stopping Condition

We calculate the objective function at each iteration for the clustering, and stop when it does not reduce by a certain set fraction (say 0.5%) for three consecutive iterations.