Appendix- 4

Iterative Closest Point Registration

Introduction

In the Iterative Closest Point, one point cloud (vertex cloud), the *reference*, or *target*, is kept fixed, while the other one, the *source*, is transformed to best match the reference. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize an error metric, usually a distance from the source to the reference point cloud, such as the sum of squared differences between the coordinates of the matched pairs. ICP is one of the widely used algorithms in aligning three dimensional models given an initial guess of the rigid body transformation required. The ICP algorithm was first introduced by Chen and Medioni, and Besl and McKay.

The Iterative Closest Point algorithm treats correspondence as a variable to be estimated.

Inputs:

reference and source point clouds, initial estimation of the transformation to align the source to the reference (optional), criteria for stopping the iterations.

Output:

refined transformation.

Algorithm Steps

1. For each point (from the whole set of vertices usually referred to as dense or a selection of pairs of vertices from each model) in the source point cloud, Match the closest point in the reference point cloud (or a selected set).
2. Estimate the combination of rotation and translation using a root mean square point to point distance metric minimization technique which will best align each source point to its match found in the previous step. This step may also involve weighting points and rejecting outliers prior to alignment.

3. Transform the source points using the obtained transformation.
4. Iterate (re-associate the points, and so on).

Zhang proposes a modified K-D tree algorithm for efficient closest point computation. In this work a statistical method based on the distance distribution is used to deal with outliers, occlusion, appearance, and disappearance, which enables subset-subset matching.

There exist many ICP variants, from which point-to-point and point-to-plane are the most popular. The latter usually performs better in structured environments.