Comparison of ICP-Algorithms

B. Glomann, X. Laboureux, S. Seeger

In order to digitize the surface of a complex object, several range images have to be taken from different viewpoints. To generate a complete model from these range images, in a first step all of them have to be transformed to the same coordinate system [Fig. **??**]. In order to find the rotation and translation between two range images of overlapping surface regions – this process is called *registration* – usually an *Iterated-Closest-Points (ICP)* algorithm is performed after a coarse alignment of the images has been given. Since the ICP algorithm allows much freedom in its implementation we compared several of such possibilities with respect to their accuracy and speed. All implementations based on triangle nets as input data.

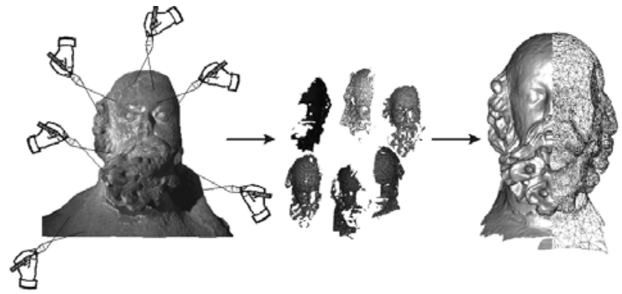

Figure 1: Range images from different viewpoints have to be transformed to the same coordinate system.

Introduction

In an ICP-algorithm, closest points between two data sets are identified as corresponding points and then a cost function depending on their distances is minimized with respect to the six rotation and translation parameters. After determining these parameters the resulting transformation is applied to the respective data set, so that both data sets come closer to each other. This process is iterated until convergence [1].

This algorithm allows much freedom in the implementation of the closest point determination, in the choice of the cost function and in the way of finding the optimal transformation in each iteration step. In addition, there are a lot of options to handle outliers in the data sets.

We implemented eight different methods to find closest points and eight different methods for the handling of outliers. Until now the only cost function we used is a least squares sum of distances of corresponding points. In addition, we implemented three different optimization strategies.

Results

We reached the best trade-off between evaluation time and accuracy by using the following methods for an ICP-algorithm: For the closest point search we used a so called k-D tree [2] in the first iteration and in all further iterations we performed only a local search over a given neighborhood of the last iteration's closest point. In addition, we do not simply take the closest point of the second data set, but the closest point on the triangles of the second data set. By using this linear interpolation between the points of the second data set we got remarkably better results with only neglectable additional time costs.

We would like to emphasize that outliers are of great importance. Boundary points or usual outliers make the registration results extremely dependent on the distance threshold that is used to reject points as corresponding points. A method for automatically finding an optimal threshold is in progress.

We realized that the fastest strategy to optimize the cost function in each iteration step is to use an approximate solution by linearizing the transformation parameters in the least squares sum and solving the resulting system of linear equations by an LU decomposition [4]. This is remarkable since it is in contradiction to [2] where it is claimed that closed form solutions[1] are significantly faster.

Final remarks

We would like to emphasize that the implemented ICP-algorithms use triangle nets as input data. In this way we are not restricted to the raster data of our 3D-sensors. Our algorithms can also be used for isosurfaces that are extracted from volume data (CT, MRI, PET, SPECT, ...) and for data produced by other 3D techniques like e.g. shape from motion.

[1] P.J. Besl, N.D. McKay, *A method for registration of 3-d shapes*, PAMI, 14(2), 1992, pp. 239 – 256
[2] Z. Zhang, *Iterative point matching for registration of free-form curves and surfaces*, IJCV, 13(1), 1994, pp. 119 – 152
[3] K.S. Arun et al., *Least-squares fitting of two 3-d point sets*, PAMI, 9(5), 1987, pp. 698 –700
[4] W.H. Press et al., *Numerical Recipes in C*, Cambridge University Press, 1996.

[1]It is remarkable that such closed form solutions (to the problem of optimizing a least squares sum of distances of corresponding points) exist.