A Framework for a Comparison of ICP-Algorithms

S. Seeger, B. Glomann, X. Laboureux

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. In order to find the rotation and translation between two range images of overlapping surface regions – this process is called *registration* – usually a variation of an *Iterated-Closest-Point (ICP)* algorithm is performed when a rough alignment of the images is already given. We developed a class library that allows the documentation of different ICP-algorithms and their – due to a common code basis – objective comparison with respect to their accuracy and speed.

ICP-algorithms in general: The general approach of an ICP-algorithm can be described as follows [1]: Given two 2.5D data sets from different views of an object, in a first step for each point in data set 1, the closest point defined by data set 2 is searched. The points of the resulting pairs are *interpreted* as *corresponding points* that differ from each other only by a rotation and translation. Since such a transformation is already overdetermined for more than three non-collinear point pairs, the transformation that is in best accordance with the given point pairs is searched as the minimum of a cost function (typically a least squares sum of distances of corresponding points). After determining the best transformation the resulting rotation and translation is applied to the respective data set, so that both data sets come closer to each other. Now again a closest point search is performed introducing the next iteration step. The process is repeated until an abort criterion is reached.

Questions in the general approach: This general description of an ICP-algorithm raises several questions: for example, how is a closest point defined in detail (unclear, since it is not defined, whether the points of data set 2 define a surface and in case, which one). In addition it remains unclear, which cost function is used for the determination of the best transformation between the two data sets. Furthermore, the strategy of determining the best transformation for a given cost function is not described. Finally, the abort criterion of the whole algorithm is not defined.

Besides these obvious questions there are some more which are not as apparent: How is the closest point determination performed in detail for a given closest point definition? In which way outliers (boundary points and usual outliers) are treated? What happens, if one uses a symmetrical strategy for the determination of corresponding points (closest points are not only determined from data set 1 to data set 2 but also the other way round)?

Our approach: All these questions cannot be uniquely answered but represent freedoms in the concrete implementation of an ICP-algorithm. To explicitly document these freedoms we developed a class library of *registration energies*: different registration energies differ in their closest point determination[1], in the choice of the cost function for the determination of the best transformation and in the different treatment of outliers. Each combination of these freedoms defines a new registration energy that is documented in a C++ class.

We chose *energy* as the name for the classes since they represent the functions that are minimized during the whole ICP-algorithm. Each combination of the above implementation freedoms results in another minimum, so that each combination has to belong to another function that is minimized.

The theoretical framework of the ICP-algorithm itself was realized with the help of the object oriented ideas of abstract base classes and abstract operations (pure virtual functions in C++).

All registration energies own an (inherited) method for finding the next best transformation (the next minimum) for a given start position (i.e. for a given rough alignment of the data sets). It is possible to give this method a strategy for the determination of the best transformation as well as an abort criterion as parameters, so that also these implementation freedoms are taken into account.

Final conclusions: Since all implemented ICP algorithms use the same code basis, the class library of registration energies is a good starting point for the objective comparison of different registration approaches with respect to their speed and accuracy. Some results of this comparison can be found in our subsequent report [2].

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 as the reverse engineering software SLIM, that was developed in our group at the Chair for Optics. Our algorithms can also be used for isosurfaces that are extracted from volume data (CT, MRI, PET, SPECT, ...) and for data produced by 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] S. Seeger, B. Glomann, X. Laboureux, *Comparison of ICP-Algorithms: First Results*, Tech. Rep.

[1] The closest point determination includes a closest point definition, an algorithm for the (approximate) finding of closest points and the type of strategy (symmetrical or asymmetrical).