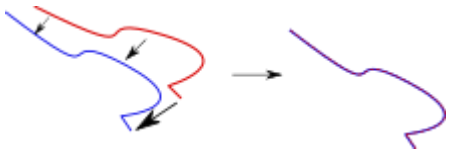
WIKIPEDIA

Iterative closest point

Iterative closest point (**ICP**)[1][2][3][4] is an algorithm employed to minimize the difference between two clouds of points. ICP is often used to reconstruct 2D or 3D surfaces from different scans, to localize robots and achieve optimal path planning (especially when wheel odometry is unreliable due to slippery terrain), to co-register bone models, etc.



Idea behind the iterative closest point algorithm

Contents

Overview

In the Iterative Closest Point or, in some sources, the Iterative Corresponding 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 transformation required.[5] The ICP algorithm was first introduced by Chen and Medioni,[3] and Besl and McKay.[2]

The Iterative Closest Point algorithm contrasts with the Kabsch algorithm and other solutions to the orthogonal Procrustes problem in that the Kabsch algorithm requires correspondence between point sets as an input, whereas Iterative Closest Point 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.

Essentially, the algorithm steps are:[5]

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 [4] 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,[6] from which point-to-point and point-to-plane are the most popular. The latter usually performs better in structured environments.[7][8]

Implementations

- MeshLab an open source mesh processing tool that includes a GNU General Public License implementation of the ICP algorithm.
- CloudCompare an open source point and model processing tool that includes an implementation of the ICP algorithm. Released under the GNU General Public License.
- PCL (Point Cloud Library) is an open-source framework for n-dimensional point clouds and 3D geometry processing. It includes several variants of the ICP algorithm.[9]
- Open source C++ implementations of the ICP algorithm are available in VTK, ITK and Open3D (http://www.open3d.org) libraries.
- libpointmatcher (https://github.com/ethz-asl/libpointmatcher) is an implementation of point-to-point and point-to-plane ICP released under a BSD license.
- simpleICP (https://github.com/pglira/simpleICP) is an implementation of a rather simple version of the ICP algorithm in various languages.

See also

- Point set registration

References

1. Arun, Somani; Thomas S. Huang; Steven D. Blostein (1987). "Least-square fitting of two 3-D point sets". *IEEE Pattern Analysis and Machine Intelligence*. **9** (5): 698–700. doi:10.1109/TPAMI.1987.4767965 (https://doi.org/10.1109%2FTPAMI.1987.4767965). PMID 21869429 (https://pubmed.ncbi.nlm.nih.gov/21869429).
2. Besl, Paul J.; N.D. McKay (1992). "A Method for Registration of 3-D Shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **14** (2): 239–256. doi:10.1109/34.121791 (https://doi.org/10.1109%2F34.121791).
3. Chen, Yang; Gerard Medioni (1991). "Object modelling by registration of multiple range images". *Image Vision Comput*. **10** (3): 145–155. doi:10.1016/0262-8856(92)90066-C (https://doi.org/10.1016%2F0262-8856%2892%2990066-C).
4. Zhang, Zhengyou (1994). "Iterative point matching for registration of free-form curves and surfaces". *International Journal of Computer Vision*. **13** (12): 119–152. CiteSeerX 10.1.1.175.770 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.175.770). doi:10.1007/BF01427149 (https://doi.org/10.1007%2FBF01427149).
5. Rusinkiewicz, Szymon; Marc Levoy (2001). *Efficient Variants of the ICP Algorithm*. Proceedings Third International Conference on 3-D Digital Imaging and Modeling. Quebec City, Quebec, Canada. pp. 145–152. doi:10.1109/IM.2001.924423 (https://doi.org/10.1109%2FIM.2001.924423).
6. Pomerleau, François; Colas, Francis; Siegwart, Roland (2015). "A Review of Point Cloud Registration Algorithms for Mobile Robotics" (https://www.researchgate.net/publication/277558596). *Foundations and Trends in Robotics*. **4** (1): 1–104. CiteSeerX 10.1.1.709.2212 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.709.2212). doi:10.1561/2300000035 (https://doi.org/10.1561%2F2300000035).

7. Kok-Lim Low (February 2004). "Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration" (http://www.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf) (PDF). *Comp.nys.edu.sg*. Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill. Retrieved 2017-02-27.
8. François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing ICP Variants on Real-World Data Sets. (http://stephane.magnenat.net/publications/Comparing%20ICP%20Variants%20on%20Real-World%20Data%20Sets%20-%20Pomerleau%20et%20al.%20-%20Autonomous%20Robots%20-%202013.pdf) In Autonomous Robots, 34(3), pages 133–148, DOI: 10.1007/s10514-013-9327-2, April 2013.
9. Holz, Dirk; Ichim, Alexandru E.; Tombari, Federico; Rusu, Radu B.; Behnke, Sven (2015). "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D" (https://www.researchgate.net/publication/283198426). *IEEE Robotics Automation Magazine*. **22** (4): 110–124. doi:10.1109/MRA.2015.2432331 (https://doi.org/10.1109%2FMRA.2015.2432331).
