64-424
Intelligent Robotics

https://tams.informatik.uni-hamburg.de/lectures/2019ws/vorlesung/ir

Marc Bestmann / Michael Görner / Jianwei Zhang

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

Winterterm 2019/2020

Outline

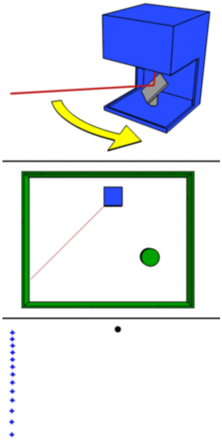
Outline

1. Scan processing
   - Scan filtering
   - Feature extraction
   - Scan Matching

Table of Contents

Motivation

- Distance scans are a common sensor input
  - Lidar
  - Depth camera
  - ...
- Often used for localization
- Large data amount
- Errors are common
- Mostly 3D scans, but we will use 2D in this lecture



LRF

GIF

Scan filtering

Several approaches to processing/filtering of distance measurement data

- Scan data filtering:
  - Smoothing
  - Data reduction
- Feature extraction:
  - Line segments
  - Corners
  - ...
- Clustering/classification

Scan filtering (cont.)

- A scan is a set of measurement values

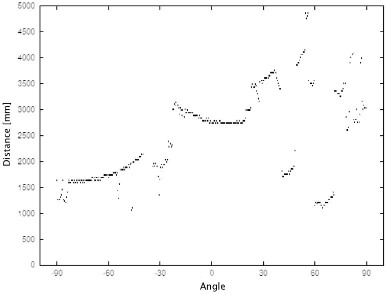
$$\left\{ m_i = (\alpha_i, r_i)^T | i = 0 \ldots n-1 \right\}$$

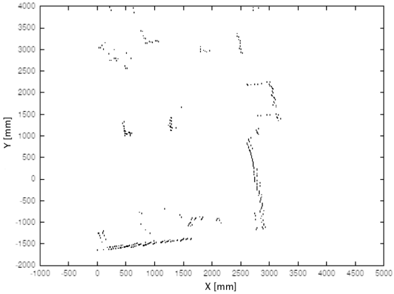
specified in polar coordinates $(\alpha_i, r_i)^T$

- For a given measuring location $p = (x, y, \theta)^T$ a scan point $m_i = (\alpha_i, r_i)^T$ can be converted to cartesian coordinates

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} r_i \cos\alpha_i \\ r_i \sin\alpha_i \end{bmatrix}$$

Scan filtering (cont.)





Scan filtering (cont.)

- **Issues**: Big amount of data, noise/outliers, etc.
- **Solution**: Application of filtering procedures according to requirements
- Basic scan data filters are:
  - Median filter
  - Reduction filter
  - Angle reduction filter

Median filter

The median filter recognizes outliers and replaces them with a more suitable measurement

- A window is placed around each scan point, containing measurements before and after the point
- The value of the scan point is replaced by the median distance within the filter window
- Window size (*wSize*) is the main parameter of the median filter
- Big window sizes lead to a strong smoothing effect
- *Disadvantage*: Corners are rounded

Median filter (cont.)

Example

Raw sensor input:

0.9	0.8	0.1	1	1.1	0.7	1

Applying median filter on fourth value.

Sort entries in window and take center value as result.

Window size 3: [0.1, **1**, 1.1]

Window size 5: [0.1, 0.7, **0.8**, 1, 1.1]

Window size 7: [0.1, 0.7, 0.8, **0.9**, 1, 1, 1.1]

Median filter (cont.)

Reduction filter

The reduction filter reduces point clusters to a single point

- A point cluster is specified through a radius r from the starting point
- The first point (starting point) of a scan starts the first cluster
- All subsequent points at a distance $d < 2 \cdot r$ are added to the cluster
- A new cluster is started at the first point with a bigger distance
- Each cluster is replaced by the center of gravity of the corresponding points

Reduction filter (cont.)

Reduction filter (cont.)

- The reduction filter algorithm has a linear time complexity
- *Advantages* of the reduction filter:
  - Reduction of the number of scan points without significant information loss
  - This leads to shorter duration of scan post-processing
  - The result is a more uniform distribution of the points
- *Disadvantages* of the reduction filter:
  - Feature extraction is not as easy any more
  - Possibly too few points for a feature (e.g. corner)

Angle reduction filter

The angle reduction filter resembles the reduction filter

- Scan points having a similar measurement angle are grouped and replaced by the point with the median distance
- The angle reduction filter is used for an even reduction of scan data that have a high angular resolution
- The time complexity of the angle reduction filter is linear

Angle reduction filter (cont.)

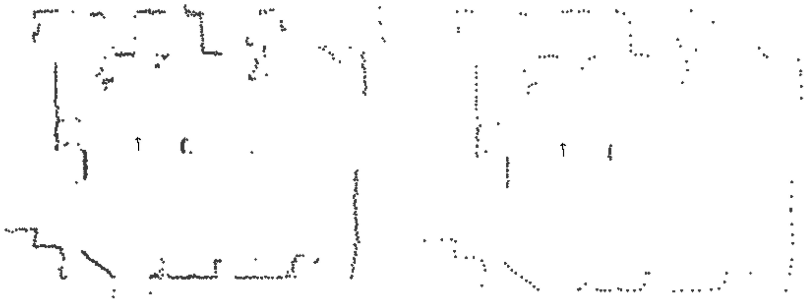


Table of Contents

Feature extraction

General approach:

- Extraction of features instead of low-level processing of complete scans
- But what are useful features in point data?
- Common features: Lines, Corners

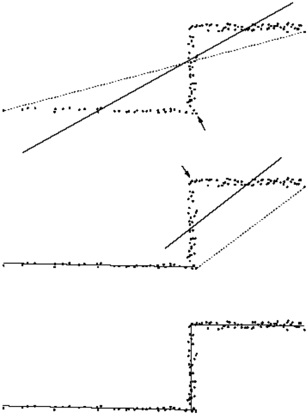
Line Detection by:

- Devide and Conquer Regression
- Hough-Transform
- RANSAC
- ...

Lines - Divide And Conquer

- Initially a regression line is fitted to the points
- If the deviation is too big, the set of points is divided
- Dividing point is the one with the highest distance to the line
- Critical parameters:
  - Minimum number of points to form a line
  - Maximum allowed deviation
- Time complexity similar to *Quicksort*: quadratic in the worst case, logarithmic on average

Lines - Divide And Conquer (cont.)



Lines - Divide And Conquer (cont.)

Hough transform

The Hough transform is a feature extraction approach applied in digital image processing

- Recognition of lines, circles, . . .
- Points in the image are mapped onto a parameter space
- Suitable parameters:
  - Line: Slope and y-intercept
  - Circle: Radius and center
- Searched figure is located at the clusters in parameter space
- Usually implemented by histogram-analysis

Straight line recognition

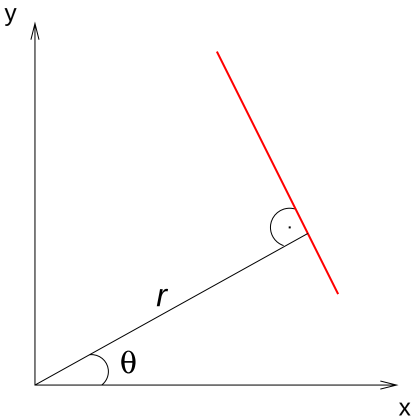
- **Parameters:** Slope and y-intercept
- **Disadvantage:** Straight lines having an infinite slope can not be mapped
- **Better:** Straight line in Hessian normal form

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

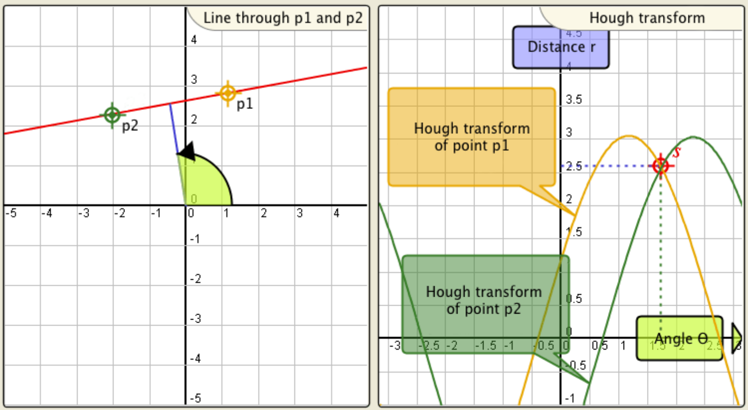
with

- θ: Angle between x-axis and normal of the straight line
- r: Distance between origin and straight line

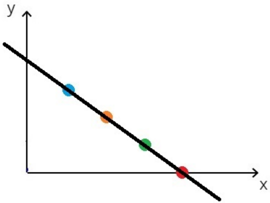
Straight line recognition (cont.)



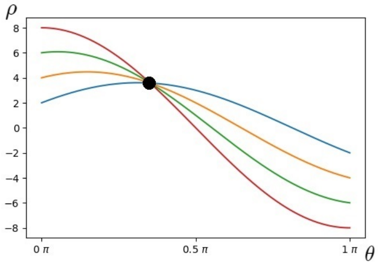
Straight line recognition (cont.)



Straight line recognition (cont.)



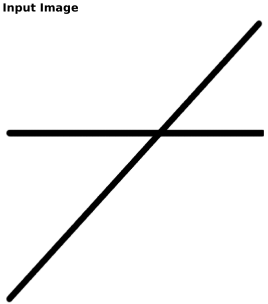
Points which form a line

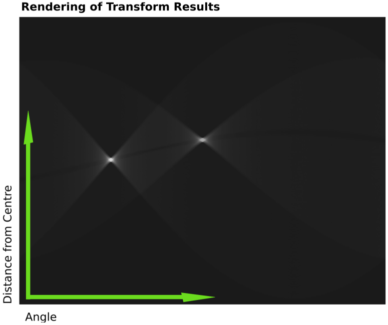


Bunch of sinusoids intersecting at one point

https://tomaszkacmajor.pl/index.php/2017/06/05/hough-lines-transform-explained/

Straight line recognition (cont.)





The axis zero point is in the center of the images

https://tomaszkacmajor.pl/index.php/2017/06/05/hough-lines-transform-explained/

Straight line recognition (cont.)

All extracted/recognized line segments can be formulated in Hessian normal form

- Each relevant scan data point is tested with several value pairs from the parameter space
- Points of intersection in parameter space represent potential parameter candidates for the straight line found in the scan data
- If multiple candidates exist *clusters* are formed
- The θ-r-point representing the parameters of the straight line is determined as the center of gravity

Table of Contents

Scan matching

In mobile robotics, scan data obtained with a laser rangefinder is frequently used to determine the location of a robot on a map

- Raw scan data is transformed into a set of features (e.g. lines)
- The *a priori* available map is searched for overlap and alignment with the extracted set of features
  $\rightarrow$ e.g. ICL - *Iterative Closest Line*
- The output is a transformation that allows to determine the location that the scan was taken at (*best alignment*)
- This procedure is called scan matching
- Scan matching can be carried out using raw scan data
  $\rightarrow$ e.g. ICP - *Iterative Closest Point*

Scan matching (cont.)

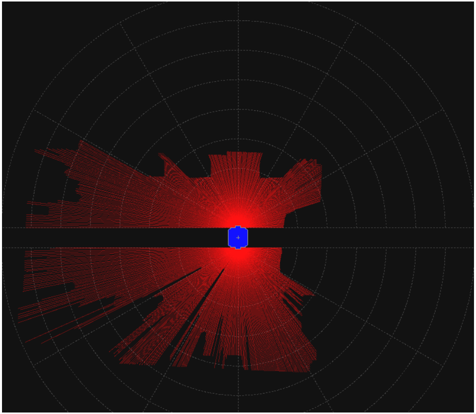
- Scan matching can be performed using:
  - Scan data and map data
  - Scan data and scan data (e.g. *previous scan*)
  - Map data and map data
- Scan matching is an optimization problem that suffers from having many local minima
- The procedure requires a rough estimate of the initial location (e.g from odometry data)
- *ICL* and *ICP* are said to converge if the initial guess is "close enough"

Scan matching (cont.)

Scan matching proceeds similar to *Expectation-Maximization algorithms*:

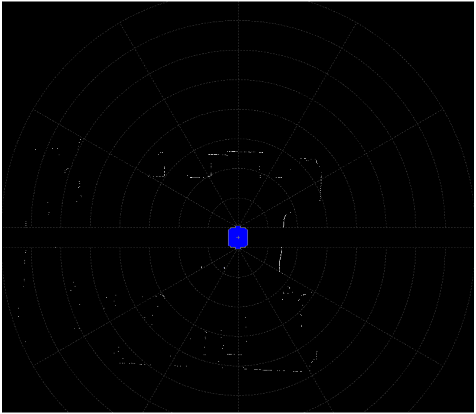
- Let $(x, y, \theta)^T = (s_x, s_y, s_\theta)^T$, where $(s_x, s_y, s_\theta)^T$ is the initial guess of the scan location based on the odometry
- Transform obtained scan data based on the initial guess $(x, y, \theta)^T$
- For each feature, determine the *target* feature closest to it
- Calculate the transformation $T = (\delta x, \delta y, \delta\theta)^T$, which minimizes the sum of squared distances between the extracted features and their targets
- Update $(x, y, \theta)^T = (x, y, \theta)^T + (\delta x, \delta y, \delta\theta)^T$
- Repeat the steps until the procedure converges

Scan matching (cont.)



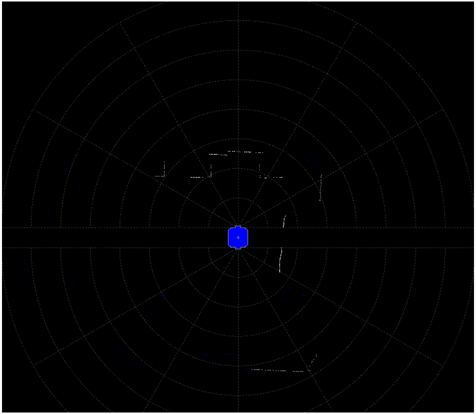
Scan data acquisition

Scan matching (cont.)



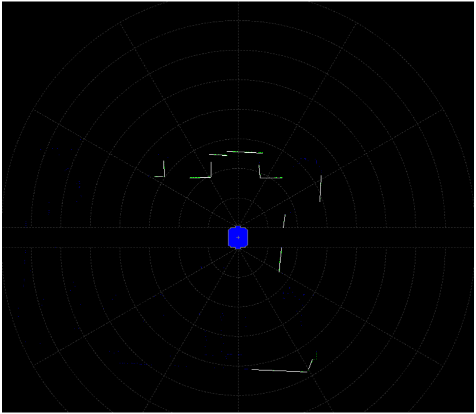
Scan conversion

Scan matching (cont.)



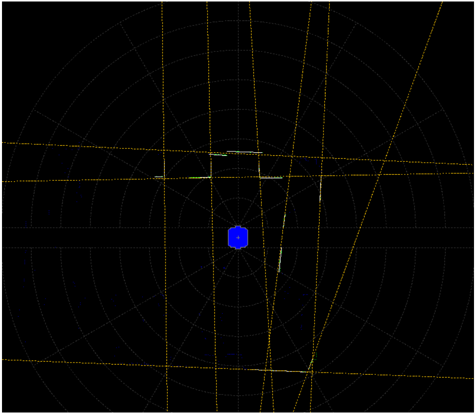
Scan filtering

Scan matching (cont.)



Feature extraction

Scan matching (cont.)

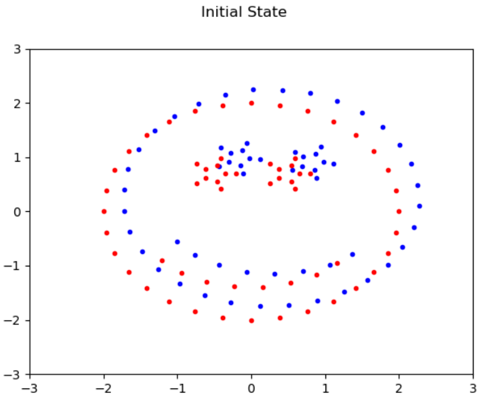


Scan matching

ICP - Iterative Closest Point

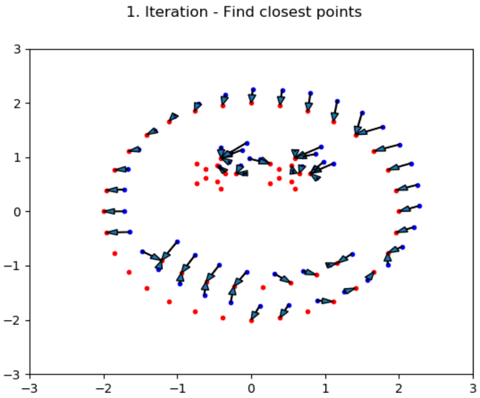
- Step 1: Match every sensed point to closest reference point
- Step 2: Compute transformation that produces least square error for point to point distance (meaning a transformation that will align the points to their partner points from step 1).
- Step 3: Transform all points with transformation from step 2
- Do this until the distance is under some threshold

ICP Algorithm



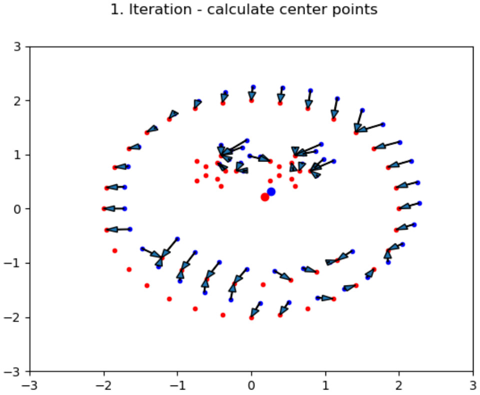
Jonas Tietz, Object reconstruction with ICP, https://tams.informatik.uni-hamburg.de/lectures/2018ws/seminar/ir/

ICP Algorithm



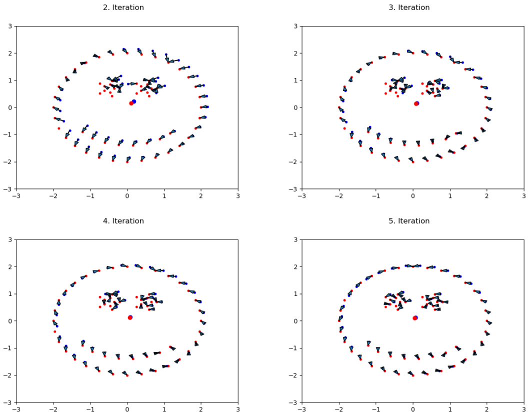
Jonas Tietz, Object reconstruction with ICP, https://tams.informatik.uni-hamburg.de/lectures/2018ws/seminar/ir/

ICP Algorithm



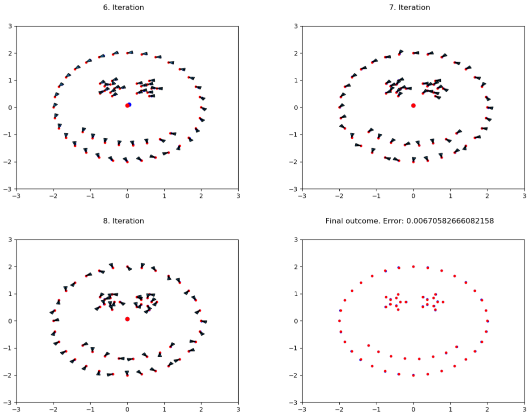
Jonas Tietz, Object reconstruction with ICP, https://tams.informatik.uni-hamburg.de/lectures/2018ws/seminar/ir/

ICP Algorithm



Jonas Tietz, Object reconstruction with ICP, https://tams.informatik.uni-hamburg.de/lectures/2018ws/seminar/ir/

ICP Algorithm



Jonas Tietz, Object reconstruction with ICP, https://tams.informatik.uni-hamburg.de/lectures/2018ws/seminar/ir/

ICP Implementation

A reference implementation can be found here:
https://github.com/AtsushiSakai/PythonRobotics/blob/master/SLAM/iterative_closest_point/iterative_closest_point.py

ICP

Video
https://www.youtube.com/watch?v=YVDHZ3Afjas

Literature list

[1] Andrea Censi.
An ICP Variant Using a Point-To-Line Metric.
In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 19–25, May 2008.

[2] Richard O. Duda and Peter E. Hart.
Use of the Hough Transformation to Detect Lines and Curves in Pictures.
Communications of the ACM, 15(1):11–15, January 1972.

Literature list (cont.)

[3] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart.
A Comparison of Line Extraction Algorithms Using 2d Laser Rangefinder for Indoor Mobile Robotics.
In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1929–1934, August 2005.