Introduction to Mobile Robotics

Iterative Closest Point Algorithm

Motivation

The Problem

- Given: two corresponding point sets:
  \[ X = \{x_1, \ldots, x_n\} \]
  \[ P = \{p_1, \ldots, p_n\} \]
- Wanted: translation \( t \) and rotation \( R \) that minimizes the sum of the squared error:
  \[ E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - Rp_i - t||^2 \]

Where \( x_i \) and \( p_i \) are corresponding points.

Key Idea

- If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form.
Center of Mass

\[ \mu_x = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad \text{and} \quad \mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i \]

are the centers of mass of the two point sets.

Idea:

- Subtract the corresponding center of mass from every point in the two point sets before calculating the transformation.
- The resulting point sets are:
  \[ X' = \{ x_i - \mu_x \} = \{ x'_i \} \quad \text{and} \]
  \[ P' = \{ p_i - \mu_p \} = \{ p'_i \} \]

SVD

**Theorem** (without proof):

If \( \text{rank}(W) = 3 \), the optimal solution of \( E(R,t) \) is unique and is given by:

\[ R = UV^T \]
\[ t = \mu_x - R\mu_p \]

The minimal value of error function at \( (R,t) \) is:

\[ E(R,t) = \sum_{i=1}^{N_p} (||x'_i||^2 + ||y'_i||^2) - 2(\sigma_1 + \sigma_2 + \sigma_3) \]

SVD

Let \( W = \sum_{i=1}^{N_p} x'_i p'_i^T \)
denote the singular value decomposition (SVD) of \( W \) by:

\[ W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T \]

where \( U, V \in \mathbb{R}^{3 \times 3} \) are unitary, and \( \sigma_1 \geq \sigma_2 \geq \sigma_3 \) are the singular values of \( W \).

ICP with Unknown Data Association

- If correct correspondences are not known, it is generally impossible to determine the optimal relative rotation/translation in one step.
**ICP-Algorithm**

- Idea: iterate to find alignment
- Iterated Closest Points (ICP) [Besl & McKay 92]
- Converges if starting positions are "close enough"

![ICP-diagram](image)

**ICP-Variants**

- Various aspects of performance:
  - Speed
  - Stability (local minima)
  - Tolerance wrt. noise and/or outliers
  - Basin of convergence (maximum initial misalignment)

**Performance of Variants**

- Here: properties of these variants
ICP Variants

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Data association
4. Rejecting certain (outlier) point pairs

Selecting Source Points

- Use all points
- Uniform sub-sampling
- Random sampling
- Feature based Sampling
- Normal-space sampling
  - Ensure that samples have normals distributed as uniformly as possible

Normal-Space Sampling

uniform sampling  normal-space sampling

Comparison

- Normal-space sampling better for mostly-smooth areas with sparse features [Rusinkiewicz et al.]

Random sampling  Normal-space sampling
**Feature-Based Sampling**

- try to find “important” points
- decrease the number of correspondences
- higher efficiency and higher accuracy
- requires preprocessing

**ICP Variants**

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Data association
4. Rejecting certain (outlier) point pairs

**Selection vs. Weighting**

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates
- Weighting strategies turned out to be dependent on the data.
- Preprocessing / run-time cost tradeoff (how to find the correct weights?)

**ICP Variants**

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Data association
4. Rejecting certain (outlier) point pairs
**Data Association**
- has greatest effect on convergence and speed
- Closest point
- Normal shooting
- Closest compatible point
- Projection
- Using kd-trees or oc-trees

**Closest-Point Matching**
- Find closest point in other the point set
- Closest-point matching generally stable, but slow and requires preprocessing

**Normal Shooting**
- Project along normal, intersect other point set
- Slightly better than closest point for smooth structures, worse for noisy or complex structures

**Point-to-Plane Error Metric**
- Using point-to-plane distance instead of point-to-point lets flat regions slide along each other [Chen & Medioni 91]
**Projection**

- Finding the closest point is the most expensive stage of the ICP algorithm
- Idea: simplified nearest neighbor search
- For range images, one can project the points according to the view-point [Blais 95]

**Projection-Based Matching**

- Slightly worse alignments per iteration
- Each iteration is one to two orders of magnitude faster than closest-point
- Requires point-to-plane error metric

**Closest Compatible Point**

- Improves the previous two variants by considering the compatibility of the points
- Compatibility can be based on normals, colors, etc.
- In the limit, degenerates to feature matching

**ICP Variants**

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Nearest neighbor search
4. Rejecting certain (outlier) point pairs
**Rejecting (outlier) point pairs**

- sorting all correspondences with respect to there error and deleting the worst t%, Trimmed ICP (TrICP) [Chetverikov et al. 2002]
- t is to Estimate with respect to the Overlap

**Problem:** Knowledge about the overlap is necessary or has to be estimated.

---

**ICP-Summary**

- ICP is a powerful algorithm for calculating the displacement between scans.
- The major problem is to determine the correct data associations.
- Given the correct data associations, the transformation can be computed efficiently using SVD.

**Application**

**Introduction to Mobile Robotics**

3D-Mapping
Urban Outdoor Scenario

- Autonomous outdoor navigation is a challenging problem
- outdoor map building
- real time path planning
- real time localization

- One of the key problems: efficient terrain representation

Typical Representations

- Collection of all 3D-points
  - ~200,000 points per scan
  - low utility for navigation
- 3D-Grid
  - huge computational and memory requirements
  - higher accuracy
- 2D-Grid
  - low cost
  - efficient for navigation
  - approximation

Alternative: Surface Map Representations

Acquisition of 3D Range Data

Updating Outdoor Maps

Kalman filter equations

\[ \hat{\mu}_t = \frac{\sigma^1_t \hat{\mu}_{t-1} + \sigma^1_{t-1} z_t}{\sigma^1_{t-1} + \sigma^1_t} \]

\[ \sigma^1_t = \frac{\sigma^1_{t-1} \sigma^1_t}{\sigma^1_{t-1} + \sigma^1_t} \]
ICP-Algorithm

- If correct correspondences between two point sets are known, it is possible to find correct relative rotation/translation in closed form.

ICP-Algorithm

- Given that $u_{i_c}$ and $u'_{j_c}$ are corresponding points.
- Try to find the parameters $R$ and $t$ which minimize the sum of the squared error $e(R,t)$

$$e(R,t) = \sum_{i=1}^{C_1} d_i(u_{i_c}, u'_{i_c}) + \sum_{j=1}^{C_2} d(v_{j_c}, v'_{j_c}) + \sum_{k=1}^{C_3} d(w_{j_c}, w'_{j_c})$$

ICP for Multi-Level Surface Maps

Learning Large Scale MLS Maps with Loops

- 35,000,000 scan points
- 135 single 3D scans collected in 2 loops
- size of learned map: 160 x 120 meters
Learning Large Scale MLS Maps with Loops in Non Flat Environments

Example of a non-flat environment

Point set with 45,139,000 3D-Points reduced to one Multi-Level Surface Map with:
- Map Size: 299m x 147m
- Cell Size: 0.1m x 0.1m
- Loop length 560m
- 73.33 Mbytes to store

Triangulated Mesh Visualization

Esplanade EPFL Lausanne

http://www.smart-team.ch
Non-Urban Outdoor Scenario

http://www.smart-team.ch

Mapping of Rough Terrain

Summary

- MLS Maps are an efficient data structure which is able to model environments with multiple levels
- Efficient matching of 3D range data using Multi-Level Surface Maps
- Traversibility and obstacle detection
- Classification of Surface Patches for:
  - fast, feature based matching via ICP
  - vertical and overhanging objects
  - multi-level-extension

Thanks for your attention