A Partitioned Approach for Efficient Graph-Based Place Recognition

Mattia G. Gollub, Renaud Dubé, Hannes Sommer, Igor Gilitschenski, Roland Siegwart
Problem

→ Processing 3D point clouds can be computationally expensive.
Problem
→ Processing 3D point clouds can be computationally expensive.

Idea
→ Recognize places on the basis of segment matching.
Why segments?

→ Good compromise between local and global descriptors.
Why segments?

→ Good compromise between local and global descriptors.

→ Do not rely on the “presence of objects” in the scene.
Why segments?

- Good compromise between local and global descriptors.
- Do not rely on the “presence of objects” in the scene.
- Do not rely on a “perfect segmentation”.
Why segments?

→ Good compromise between local and global descriptors.
→ Do not rely on the “presence of objects” in the scene.
→ Do not rely on a “perfect segmentation”.
→ Allow for descriptive and compact map representation.
**SegMatch**

- **Ground removal + Euclidean segmentation.**

**Autonomous Systems Lab**
Eigen value based features [1].

(1) k-NN retrieval.
(2) Random forest classifier trained on separate data.
Simple descriptors $\rightarrow$ High fraction of false correspondences.

Geometric consistency grouping method [2].

Geometric consistency grouping [2]
Geometric consistency grouping [2]

→ Find the largest group of pairwise geometrically consistent correspondences.

\[ |d_l (c_i, c_j) - d_t (c_i, c_j)| \leq \epsilon \]

Geometric consistency grouping [2]

→ Find the largest group of pairwise geometrically consistent correspondences.

\[ \left| d_l (c_i, c_j) - d_t (c_i, c_j) \right| \leq \epsilon \]

Method:

1. For each correspondence, initialize a new group.

Geometric consistency grouping [2]

→ Find the largest group of pairwise geometrically consistent correspondences.

\[ |d_l (c_i, c_j) - d_t (c_i, c_j)| \leq \epsilon \]

Method:

1. For each correspondence, initialize a new group.
2. For each group, iterate over all the other correspondences. Add the correspondence to the group if it is consistent with all the elements in the group.

Geometric consistency grouping [2]

→ Find the largest group of pairwise geometrically consistent correspondences.

\[ |d_l(c_i, c_j) - d_t(c_i, c_j)| \leq \epsilon \]

Method:

1. For each correspondence, initialize a new group.
2. For each group, iterate over all the other correspondences. Add the correspondence to the group if it is consistent with all the elements in the group.
3. Select the biggest group and obtain the localization transformation with RANSAC.


Autonomous Systems Lab
Geometric consistency grouping [2]

→ Worst case asymptotic complexity $O(n^3)$

→ Can find a suboptimal solution depending on vertices ordering.
Graph-based recognition

- Problem represented as a consistency graph:
  - Correspondences $\rightarrow$ Vertices
  - Consistencies $\rightarrow$ Edges
Graph-based recognition

- Problem represented as a consistency graph:
  - Correspondences $\rightarrow$ Vertices
  - Consistencies $\rightarrow$ Edges

- Solved by maximum clique detection.
Graph-based recognition

- Problem represented as a consistency graph:
  - Correspondences $\rightarrow$ Vertices
  - Consistencies $\rightarrow$ Edges

- Solved by maximum clique detection.

- Identify transformation by least squares (Umeyama method).
Graph-based recognition

- Problem represented as a consistency graph:
  - Correspondences $\rightarrow$ Vertices
  - Consistencies $\rightarrow$ Edges

Naïve graph construction $O(n^2)$

- Solved by maximum clique detection.

- Identify transformation by least squares (Umeyama method).
Graph-based recognition

- Problem represented as a consistency graph:
  - Correspondences $\rightarrow$ Vertices
  - Consistencies $\rightarrow$ Edges

Naïve graph construction $O(n^2)$

- Solved by maximum clique detection.
  $\rightarrow$ Generally NP-complete

- Identify transformation by least squares (Umeyama method).
Partition-based graph construction
Partition-based graph construction

Observation: Two consistent correspondences must follow

\[ d_t(c_i, c_j) \leq b + \epsilon \]
Partition-based graph construction

Observation: Two consistent correspondences must follow

\[ d_t(c_i, c_j) \leq b + \epsilon \]
Partition-based graph construction
Maximum clique detection

→ We take advantage of the sparseness of the graph.

Maximum clique detection

→ We take advantage of the sparseness of the graph.

→ Search for maximum clique as proposed by Eppstein et al. [3].

Results

![Bar chart showing mean runtime (ms) for KITTI Localization and Loop Closure with PCL and Our method.]
## Results

<table>
<thead>
<tr>
<th>Step</th>
<th>Mean runtime (Localization)</th>
<th>Mean runtime (Loop-Closure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning</td>
<td>$0.06 \pm 0.01 ms$</td>
<td>$0.03 \pm 0.02 ms$</td>
</tr>
<tr>
<td>Graph construction</td>
<td>$13.86 \pm 7.95 ms$</td>
<td>$8.09 \pm 9.70 ms$</td>
</tr>
<tr>
<td>Max clique detection</td>
<td>$0.13 \pm 0.08 ms$</td>
<td>$0.17 \pm 0.23 ms$</td>
</tr>
<tr>
<td>Transformation estimation</td>
<td>$&lt; 0.01 ms$</td>
<td>$&lt; 0.01 ms$</td>
</tr>
<tr>
<td>Total (Our method)</td>
<td>$14.33 \pm 8.16 ms$</td>
<td>$8.52 \pm 9.98 ms$</td>
</tr>
<tr>
<td>Total (PCL)</td>
<td>$94.23 \pm 56.28 ms$</td>
<td>$17.49 \pm 32.10 ms$</td>
</tr>
<tr>
<td>Speedup</td>
<td>$6.57x$</td>
<td>$2.05x$</td>
</tr>
</tbody>
</table>
Thank you!

https://github.com/ethz-asl/segmatch

IROS SLAM 1 Session  MoBT7.2

mattia.gollub@hotmail.ch

rdube@ethz.ch