Fast Global Pointcloud Registration via Smart Indexing

Nicolas Mellado\textsuperscript{1}, Dror Aiger\textsuperscript{2}, Niloy J. Mitra\textsuperscript{1}

\textsuperscript{1}University College London \quad \textsuperscript{2}Google Inc.
Problem Statement

- Estimate rigid transformation $\text{tr}$
  - Large search space (6DoF)
Problem Statement

• Estimate rigid transformation $\mathbf{tr}$
  • Local registration: from an input pose
    • ICP [BM92], [CM92], [RL01], [MGPG04]
    • Sparse ICP [BTP13]
    • Kinect Fusion [IKH*11]

$\mathbf{P} + \mathbf{tr}(\mathbf{Q})$
Problem Statement

• Estimate rigid transformation $\mathbf{t}_R$
  • Local registration: from an input pose
    • ICP [BM92], [CM92], [RL01], [MGPG04]
    • Sparse ICP [BTP13]
    • Kinect Fusion [IKH*11]
  • Global registration: arbitrary input pose
    • RANSAC [FB81], [IR96], [CH99]
    • and variants [GMGP05], [PB09], [PB11], [ART10], [RABT13]
    • 4 Point Congruent Set [AMCO08]

3 pairs of corresponding points are sufficient to define a rigid transformation
Problem Statement

RANSAC: $O(n^3)$

4 Point Congruent Set (4PCS): $O(n^2)$
Problem Statement

RANSAC: $O(n^3)$

4 Point Congruent Set (4PCS): $O(n^2)$

Our approach (Super 4PCS): $O(n)$

Difficult cases
Use Super4PCS
Work with 226 points
norm_max_dist: 5.000000
Initial LCP: 0.061947
Computation time (sec): 9.996068
Score: 0.451327

(Homogeneous) Transformation from input2.obj to input1.obj:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.978</td>
<td>-0.171</td>
<td>-0.118</td>
</tr>
<tr>
<td>0.071</td>
<td>0.808</td>
<td>-0.585</td>
</tr>
<tr>
<td>0.195</td>
<td>0.564</td>
<td>0.803</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

saving transform matrix to output_transform.mx
Merged object was written to output.obj

Demo
Overview

• 4PCS
• Super4PCS
• Results
4 Point Congruent Set

- Use planar 4-points basis in \( P \)
- Find congruent 4-points in \( Q \)

4-points Congruent Sets for Robust Surface Registration
Dror Aiger, Niloy J. Mitra, Daniel Cohen-Or
SIGGRAPH 2008
4 Point Congruent Set

• What does congruent mean?

• Similar under a given transformation class
  • Ratios $r_1$ and $r_2$

$$r_1 = \frac{||a - e||}{||a - b||}$$
4 Point Congruent Set

• What does congruent mean?

• Similar under a given transformation class
  • Ratios \( r_1 \) and \( r_2 \)
  • Distances \( d_1 \), \( d_2 \)
4 Point Congruent Set

- What does congruent mean?

- Similar under a given transformation class
  - Ratios $r_1$ and $r_2$
  - Distances $d_1, d_2$
  - Angle $\alpha$

Rigid transformation
4 Point Congruent Set

Select Base

Extract pairs

Extract congruent superset

Filter congruent set

Verify and update transformation

Ratios
Distances

Filter congruent set

Ratios
Distances
Angles

False positives (non congruents)
4 Point Congruent Set

- Extract pairs
4 Point Congruent Set

- Extract pairs

\[ d_1 \approx d_1 \approx d_1 \]

\[ \varepsilon \quad d_1 \quad \varepsilon \]
4 Point Congruent Set

- Extract pairs

\[ \mathbf{e}_1 = \mathbf{a} + r_1 (\mathbf{b} - \mathbf{a}) \]

\[ \mathbf{e}_2 = \mathbf{a} + r_2 (\mathbf{b} - \mathbf{a}) \]
4 Point Congruent Set

- Extract pairs
4 Point Congruent Set

- Extract pairs
4 Point Congruent Set

- Extract pairs
- Extract congruent super-set
4 Point Congruent Set

- Extract pairs
- Extract congruent super-set
4 Point Congruent Set

- Extract pairs
- Extract congruent super-set
- Filter congruent set
4 Point Congruent Set

Select Base

Extract pairs

Extract congruent superset

Filter congruent set

Verify and update transformation

\( O(n^2) \)

\( O(m \log(m)) \)

\( O(k) \)

Ratios
Distances

Ratios
Distances

Angles

False positives (non congruents)

\( n \) points

\( m \) pairs

\( k \) congruent sets
Super 4PCS

- $O(n^2)$
  - $O(n)$
  - Verify and update transformation

- $O(m\log(m) + k)$
  - $O(m+k)$

Select Base
Extract pairs
Extract congruent set

No more false positive

$n$ points
$m$ pairs
$k$ congruent sets
Super 4PCS

\[ O(n^2) \quad O(m \log(m) + k) \]

\[ O(n) \quad O(m+k) \]

Our smart indexing techniques produce the **same** congruent set as 4PCS but in linear time.
Super 4PCS

- $\mathcal{O}(n^2)$
  - $O(n)$
  - $O(n^2)$
  - $O(m\log(m) + k)$

Select Base

Extract pairs

Extract congruent set

Verify and update transformation

$n$ points
$m$ pairs
$k$ congruent sets
Pair extraction

• Reporting incidences: all valid pairs generated from a sphere
Pair extraction

- Reporting incidences using sphere rasterization
  - Complexity depends only on \( \epsilon \) and \( d_1 \)
Pair extraction

- Reporting incidences using sphere rasterization
- With an adaptive grid

Note: using a pre-computed tree is not optimal
Pair extraction

- Reporting incidences using sphere rasterization
- With an adaptative grid
- Simultaneously for all points
Pair extraction

• Reporting incidences using sphere rasterization
• With an adaptative grid
• Simultaneously for all points
• Theoretical complexity: $O(n)$
  (see details in the paper)
Pair extraction

- Reporting incidences using sphere rasterization
- With an adaptative grid
- Simultaneously for all points
- Theoretical complexity: $O(n)$
- In practice
  - Runtime: linear
  - Minimal memory overhead

![Graph showing time in seconds for different samples]
4PCS
Congruent set extraction

- Original approach
  - Represent a pair by 2 invariants
4PCS
Congruent set extraction

- Original approach
  - Represent a pair by 2 invariants
  - Use \textit{kd-tree} to find closest invariants

\[ O(m \log(m) \quad k \text{ congruent sets} ) \]
4PCS
Congruent set extraction

- Original approach
  - Represent a pair by 2 invariants
  - Use \textbf{kd-tree} to find closest invariants
  - Filter non congruent quadriplets

\[ O(m \log(m) + k) \]
Congruent set extraction

- Efficient indexing
  - Represent pairs as invariant + direction
Congruent set extraction

- Efficient indexing
  - Represent pairs as invariant + direction
  - Hash pairs by position and direction
Congruent set extraction

- Efficient indexing
  - Represent pairs as invariant + direction
  - Hash pairs by position and direction
- Query
  - Hash positions (closest invariants)
Congruent set extraction

- Efficient indexing
  - Represent pairs as invariant + direction
  - Hash pairs by position and direction
- Query
  - Hash positions (closest invariants)
Congruent set extraction

- **Efficient indexing**
  - Represent pairs as invariant + direction
  - Hash pairs by position and direction
- **Query**
  - Hash positions (closest invariants)
  - Theoretical complexity: $O(n)$
    
    
    \[
    \text{(see details in the paper)}
    \]
Congruent set extraction

- Efficient indexing
  - Represent pairs as invariant + direction
  - Hash pairs by position and direction

- Query
  - Hash positions (closest invariants)
  - Theoretical complexity: \( O(n) \)
  - In practice
    - Runtime: linear
    - Memory overhead: similar to kd-tree
Results
Outliers

Before ICP

After ICP

Input 1.4 sec 15 sec 30 sec

35k points
Low overlap

(no ICP)
32k points
## Low overlap

### Other examples

<table>
<thead>
<tr>
<th>Model</th>
<th>Points (x1000)</th>
<th>Overlap (%)</th>
<th>4PCS (in sec)</th>
<th>Super 4PCS (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bubba</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>2.5</td>
</tr>
<tr>
<td>Buddha</td>
<td>37</td>
<td>20</td>
<td>63</td>
<td>37</td>
</tr>
<tr>
<td>Hippo</td>
<td>32</td>
<td>25</td>
<td>11</td>
<td>0.5</td>
</tr>
</tbody>
</table>

![10k points](image1.png) ![37k points](image2.png) ![32k points](image3.png)
Low overlap + featureless

2,500 points
Multi-modal models

- **P**: 2.5M points, multiview stereo
- **Q**: 2.5M points, LIDAR
Range Images alignment

Top: Input  
Bottom: initial pose

SUPER 4PCS, no ICP
Top: 0.5 sec  
Bottom: 11 sec

ICP
Top: 16 sec  
Bottom: 28 sec

Sparse-ICP ($p = 1$)
Top: 12 sec  
Bottom: 43 sec

Sparse-ICP ($p = 0.5$)
Top: 71 sec  
Bottom: 60 sec
Kinect scans

- Chaining pairwise registration (5/6 frames)
Limitations

• Sampling sensitivity
  • Region Of Interest (ROI)
  • Cannot match *between* the points

• Metric

Select Base  Extract pairs  Extract congruent set  Verify
Conclusion

• Global matching algorithm
  • Running in linear time
  • Unstructured point clouds without normals
  • Can be combined with local descriptors

• Future work
  • Real-time using GPGPU programming
  • Alternative to Kinect Fusion
Thank you for your attention

• Super 4PCS
  • Global matching running in linear time

• Acknowledgements
  • Feedback&discussion: Duygu Ceylan, Aron Monszpart
  • SparseICP data and comparisons: Sofien Bouaziz, Andrea Tagliasacchi
  • Pisa dataset: Matteo Dellepiane

• Funding
  • Marie Curie Career Integration Grant
  • ERC Starting Grant SmartGeometry
  • Adobe Research

Code and data goo.gl/uQrhJU
github.com/smartgeometry-ucl/Super4PCS

Features:
• C++
• Based on Eigen
• Structures implemented in arbitrary dimensions