4-Points Congruent Sets for Robust Pairwise Surface Registration

Dror Aiger
Niloy J. Mitra
Daniel Cohen-Or

Ben Gurion University
IIT Delhi / TU Vienna
Tel Aviv University
4-Points Congruent Sets for Robust Pairwise Surface Registration

Niloy J. Mitra
IIT Delhi / TU Vienna

Daniel Cohen-Or
Tel Aviv University
Problem Statement

Given: Two models
Problem Statement

Goal: *Automatically* align the models
Why is this Hard?

Given: Two scans
- corrupted with *noise* and *outliers*
- in *arbitrary initial poses* with *unknown overlap*
Alignment Approach

P

Q
Alignment Approach
Alignment Approach
Alignment Approach

4-Points Congruent Sets
Alignment Approach
Alignment Approach

guess and verify

P Q
Partial Matching

4-Points Congruent Sets
Improvements

• Naïve approach → too expensive

• RANSAC
  – test only a few (correspondences) guesses
  – keep the best transform

• Look only at interesting points
Aligning with Feature Points (FP)
Aligning with Feature Points
Aligning with Feature Points
Noise, Outliers, and Holes

• Feature points \(\rightarrow\) differential entities
  – \textit{noise} + \textit{outliers} unstable FPs
    \textit{wrong} correspondence
  – integral invariants
    • overlapping regions
    • \textit{missing} data

• Noise + outliers + holes \(\rightarrow\) FPs fail

[Image: Diagram showing a point \(p\) on a curve with a radius \(r\) and a plane \(p + rB\).]

\[\text{[Pottmann et al. `07]}\]
Why not Denoise Scans?

- de-noise, compute FP-s, align
- align with 4PCS, de-noise

4-Points Congruent Sets
## Related Works

### Random Sampling:
- [Fisclar and Bolles \`81]
- [Ballard \`87]
- [Wolfson and Rigoutsos \`97]
- [Gal and Cohen-Or \`06]
- [Mitra et al. \`06]

### Feature Based Alignment:
- [Johnson \`97]
- [Mori et al. \`05]
- [Gelfand et al. \`05]
- [Guskov et al. \`05]
- [Pottmann et al. \`07]

### Non-Rigid Alignment:
- [Pauly et al. \`05]
- [Brown et al. \`07]
- SGP \`08

### Computational Geometry:
- [Huttenlocher and Ullman \`90]
- [Huttenlocher \`91]
- [Goodrich et al. \`94]
- [Irani and Raghavan \`96]
- [Chen et al. \`99]
- [Indyk et al. \`99]
- [Agarwal and Sharir \`02]
Key Observation

A pair of triples (from \( P \) and \( Q \)) is enough to uniquely define a \textit{rigid transform} \( \longrightarrow O(n^3) \)

Surprisingly, a \textit{special} set of 4-points, \textit{congruent sets}, makes the problem simpler \( \longrightarrow O(n^2) \)
4-Points Congruent Sets

• Few matches → output sensitive algorithm

• Can be efficiently extracted
4PCS Algorithm

\[ P \]

\[ Q \]
4PCS Algorithm

SelectCoplanarBase

P

Q
4PCS Algorithm

SelectCoplanarBase

P

B₁

Q
4PCS Algorithm

SelectCoplanarBase

B₁

FindCongruent

U = \{U₁, U₂, \ldots\}
4PCS Algorithm

SelectCoplanarBase

B₁

FindCongruent

U₁

4-Points Congruent Sets
4PCS Algorithm

SelectCoplanarBase

P \rightarrow B_1 \rightarrow \text{FindCongruent} \rightarrow U_2

Q
4PCS Algorithm

SelectCoplanarBase

FindCongruent

P → B₁ → U₃

Q
4PCS Algorithm

SelectCoplanarBase

B₁

FindCongruent

U₄
4PCS Algorithm

SelectCoplanarBase

B₁

FindCongruent

P

Q

retain the best transform (T₁)

U_j
4PCS Algorithm

SelectCoplanarBase

B_i

FindCongruent

U_j

T_i

RANSAC iterations

4-Points Congruent Sets
4PCS Algorithm

P

SelectCoplanarBase

B_i

FindCongruent

Q

U_j

T_i

RANSAC iterations
Random Sampling

- SelectCoplanarBase
- Bi
- FindCongruent
- Uj
- verification
- Ti

RANSAC iterations

4-Points Congruent Sets
FindCongruent
Key Observation

A pair of triples (from $P$ and $Q$) is enough to uniquely define a **rigid transform** $O(n^3)$

Surprisingly, a *special* set of 4-points, **congruent sets**, makes the problem simpler $O(n^2)$
Affine Invariance
Affine Invariance

4-Points Congruent Sets
Affine Invariance

4 coplanar points

4-Points Congruent Sets
Affine Invariance
Affine Invariance

\[ r_1 = \frac{||a - e||}{||a - b||} \]
\[ r_2 = \frac{||c - e||}{||c - d||} \]
Affine Invariance
Affine Invariance
Affine Invariance
Affine Invariance

\[ \frac{\|a' - e'\|}{\|a' - b'\|} = r_1 \]
\[ \frac{\|c' - e'\|}{\|c' - d'\|} = r_2 \]
Extracting Congruent 4-points

\{a, b, c, d\} \rightarrow e, r_1, r_2

\begin{align*}
r_1 &= \frac{\|a-e\|}{\|a-b\|} \\
\end{align*}

\begin{align*}
a', b', r_1 &\rightarrow e' \\
r_1 &= \frac{\|a'-e'\|}{\|a'-b'\|} \\
\end{align*}
Extracting Congruent 4-points

\[ r_1 = \frac{|a-e|}{|a-b|} \]

\[ r_2 = \frac{|c-e|}{|c-d|} \]
Extracting Congruent 4-points

\[ r_1 = \frac{||a-e||}{||a-b||} \]

\[ r_2 = \frac{||c-e||}{||c-d||} \]
Extracting Congruent 4-points

\[ r_1 = \frac{|a-e|}{|a-b|} \]

\[ r_2 = \frac{|c-e|}{|c-d|} \]
Extracting Congruent 4-points

\[ r_1 = \frac{\|a-e\|}{\|a-b\|} \]

\[ r_2 = \frac{\|c-e\|}{\|c-d\|} \]
What if $e_1 \neq e_2$?

$r_1 = \frac{||a-e||}{||a-b||}$

$r_2 = \frac{||c-e||}{||c-d||}$

typical scenario
What if \( e_1 = e_2 \)?

\[
\begin{align*}
    r_1 &= \frac{\|a-e\|}{\|a-b\|} \\
    r_2 &= \frac{\|c-e\|}{\|c-d\|}
\end{align*}
\]

Typical scenario:

Congruent 4-points:

\( e_1 = e_2 \)
Extracting Congruent 4-points

\[ q_1, q_2, q_3, q_4, q_5 \]
Extracting Congruent 4-points

4-Points Congruent Sets
Extracting Congruent 4-points
Extracting Congruent 4-points

4-Points Congruent Sets
Extracting Congruent 4-points

\{a, b, c, d\} \equiv \{q_1, q_2, q_3, q_4\}
FindCongruent

- For all the points arising using \( r_1 \), build an *approximate range-tree* (ANN)

- For all the points due to \( r_2 \), *quickly* lookup for neighbors using range-tree
Rigid Transformation

• Euclidean distances are preserved

\[(q_1, q_2) \rightarrow \|q_1 - q_2\| \approx \|a - b\|\]
SelectCoplanarBase
4PCS Algorithm

SelectCoplanarBase → Bi

FindCongruent → U₃

P → Tᵢ

Q

RANSAC iterations

4-Points Congruent Sets
SelectCoplanarBase

• Select 3 points (from P) at random → 4\textsuperscript{th} point to ensure (approx) coplanarity

• Overlap amount → decreasing guesses f = 1, 0.5, 0.25, ...
Results
Effect of Noise

\[ \sigma = 0.5 \]

\[ \sigma = 2.0 \]

\[ \sigma = 4.0 \]
Why not Denoise Scans?

de-noise, compute FP-s, align

align with 4PCS, de-noise
Effect of Outliers

10 %  20 %  40 %
Varying Overlap

80 %  70 %  60 %  50 %  40 %
Varying Overlap

80 %  70 %  60 %  50 %  40 %

4-Points Congruent Sets
Building Facade

4-Points Congruent Sets
Building Facade
Building Facade
Jerusalem Scan

4-Points Congruent Sets
Jerusalem Scan
Try It

http://graphics.stanford.edu/~niloy/research/fpcs/fpcs_sig_08.html
Comments

• 4PCS orthogonal to:
  – Feature Points $O(n^2) \rightarrow O(n)$
  – Local Refinement (ICP)

• Extension to affine transforms
Affine Alignment
Limitations
Acknowledgements

Funding sources:
Frankel Foundation
Austrian Science Fund FWF
Israeli Ministry of Science

Data sources:
Alexander Bogomjakov
Filip Sadlo
Baoquan Chen
Thibaut Weise

Comments:
Ran Gal
Adi Levin
Yaron Lipman
Andrei Sharf
anonymous reviewers