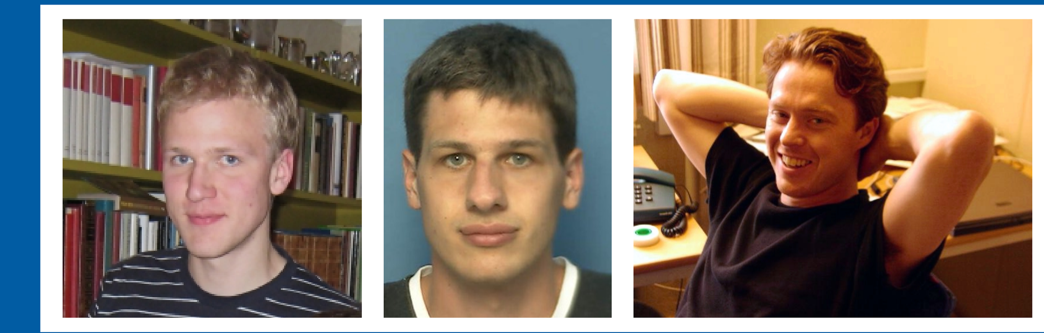
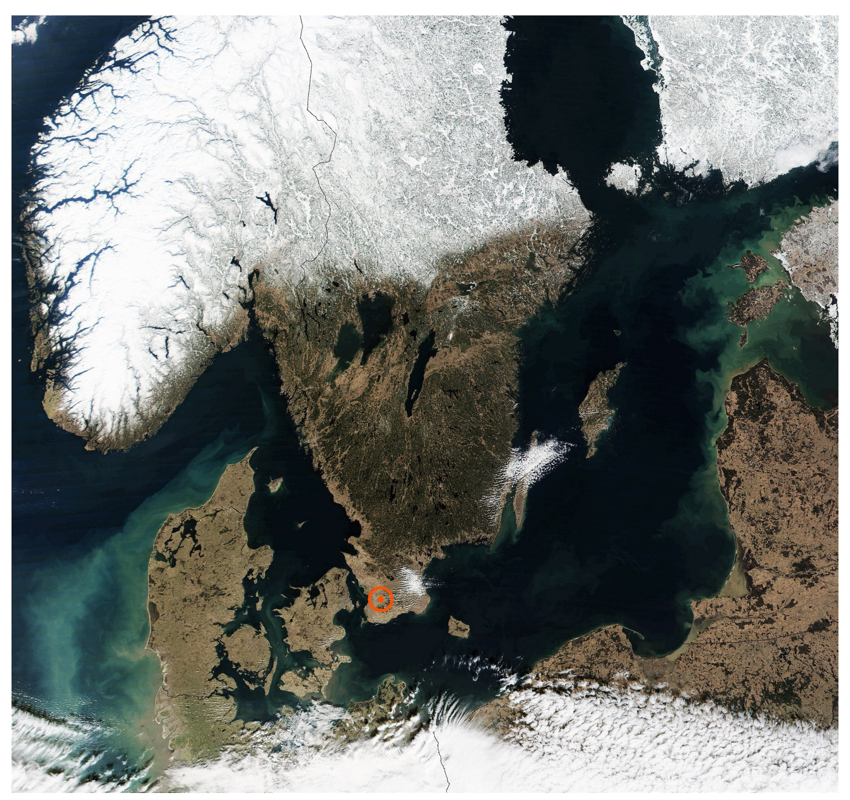
Optimal Correspondences from Pairwise Constraints

Olof Enqvist, Klas Josephson and Fredrik Kahl
Centre for Mathematical Sciences, Lund University, Sweden





Overview

We present a method to determine optimal solutions to geometric correspondence problems. Using pairwise constraints in combination with graph methods we show how to efficiently find optimal correspondences.

- Handles large rates of outliers.
- Competetive run times.

Problem Formulation

Given two point sets, find a transformation mapping one set to the other.

- An optimization approach:

Problem 1. *For a prescribed threshold $\varepsilon > 0$, find the transformation T that maximizes the number of correspondences (x_i, y_j) such that*

$$d(y_j, T(x_i)) \le \varepsilon\,.$$

Applications:

- 3D-3D registration with Euclidean errors.
- Camera pose with reprojection errors.

Pairwise Constraints

Consider a pair of correspondences. They are inconsistent if

- They match the same point.
- They are geometrically inconsistent with errors $< \epsilon$.

Gives necessary conditions for Problem 1. Sufficiency discussed in paper.

Vertex Cover

Definition 1. *A* vertex cover *for an undirected graph, is a subset S of its vertices such that every edge of the graph has at least one endpoint in S.*

Problem 2. *The* vertex cover problem *is the problem of finding the smallest vertex cover for a given graph.*

- Build a graph with one vertex per correspondence.
- Edges to mark pairwise inconsistencies.
- Remove vertices to cover all edges.

This is the minimal vertex cover problem.

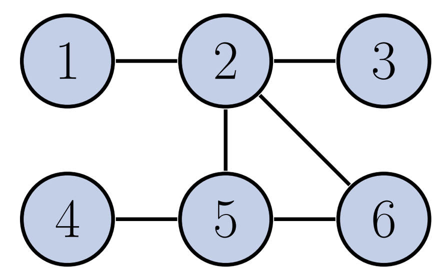


Fig. 1: Example graph for vertex cover. Minimal vertex cover: $\{2, 5\}$

- Vertex cover is NP-hard but many good algorithms exist.
- The instances from registration are often simple.

3D-3D Registration

Check point-to-point distances.

- Correspondences are inconsistent if they match close points to distant points.

Algorithm 1. *(3D-3D Registration)*

1. *Compute all pairwise distances.*
2. *Solve the vertex cover problem.*
3. *Compute the transformation for the obtained inlier set.*
4. *Verify consistency.*

2D-3D Registration

- Branch and bound over camera positions as in [EK08].
- Angles between point-camera-point are used to detect inconsistency.

Algorithm 2. *2D-3D Registration*

Initialize to get a bounded set in $\mathbb{R}^3$.
Iterate until desired precision is reached:

1. *Pick a box from the queue.*
2. *Set up a graph with all pairwise consistencies for this box.*
3. *Use the vertex cover techniques to discard the box.*
4. *If the box cannot be discarded:*
   - *Use vertex cover techniques to remove outliers*
   - *Divide the box and update the queue.*
   - *Try to update the bound on the optimum.*
5. *Remove the box from the queue.*

Results

- Simple C implementation.
- Timings for a 3.0 GHz Intel DualCore with 3GB RAM.

3D-3D Registration

Stanford Bunny

- Registration of two 500-point models.
- 250,000 hypothetical correspondences.
- Mean computation time 0.77s.

Stereo Experiments

- We built 3D models from stereo images.
- 3D-3D registration to concatenate two models.
- On average 700 correspondences, 80-90% outliers.
- Mean computation time 0.03s.

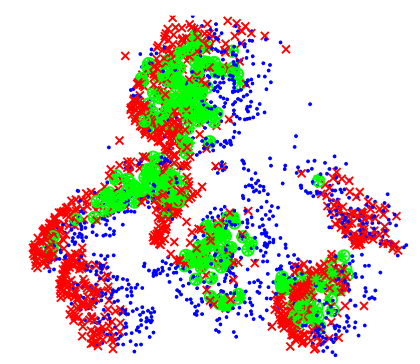


Fig. 2: 3D-3D registration of two 3D models. Green circles and stars are matched 3D points in a maximal inlier set. Red x-marks and blue dots are unmatched points in the two models.

2D-3D Registration

Lilla Fiskaregatan

About 100 images of a shopping street in Lund.

- A 3D model was built using standard techniques.
- For each view 100 points were used to estimate pose.

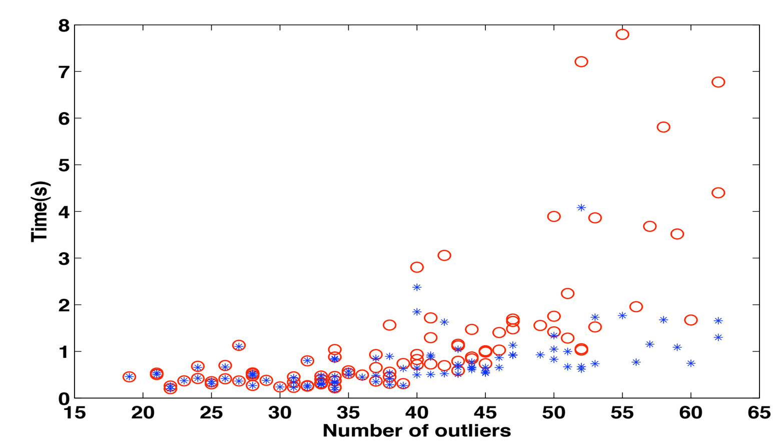
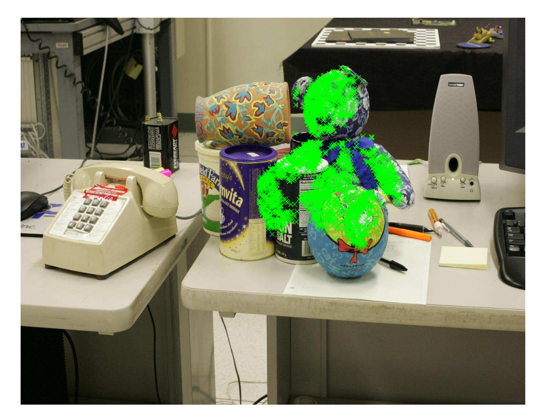


Fig. 3: Comparison with the method from [EK08]. Red rings show execution times of [EK08] and blue stars show our result.

Recognition



- Data from the stereo experiments.
- Pose algorithm as recognition tool.

Fig. 4: A bear with the 3D model reprojected onto the image.

Enforcing orientation consistency

- Include a direction to every point in the 3D model.
- Force the direction to be consistent between pairs of correspondences.
- Use SIFT to find the directions.

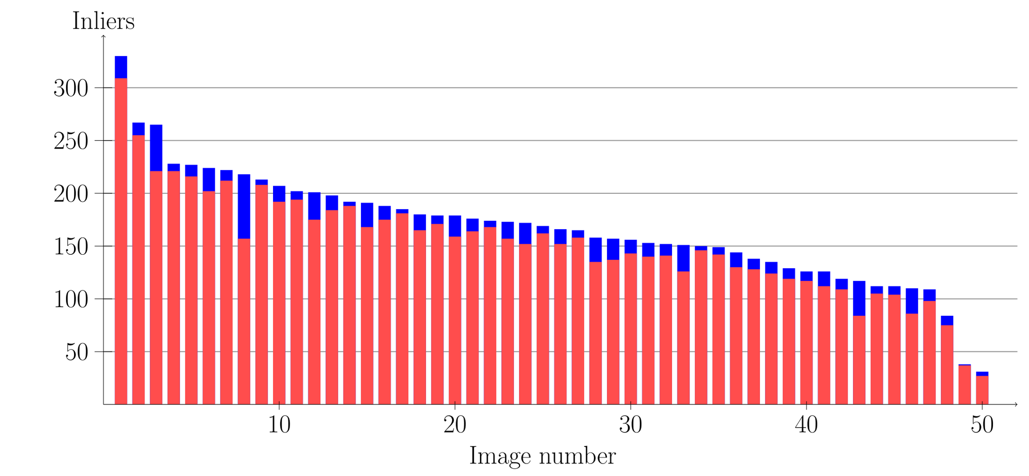


Fig. 5: Blue bars give the number of inliers when the angle constraint is not used. The red bars give the number of inliers for the same problem when the angle constraint is added.

Whiteboard

References

[BC82] R.C. Bolles and R.A. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. Journal Robotics Research*, 1(3):57–82, 1982.

[EK08] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *European Conf. Computer Vision*, pages 141–153, Marseille, France, 2008.

International Conference on Computer Vision, Kyoto, 2009