4-POINT CONGRUENT SET REGISTRATION

TORE BERGEBAKKEN

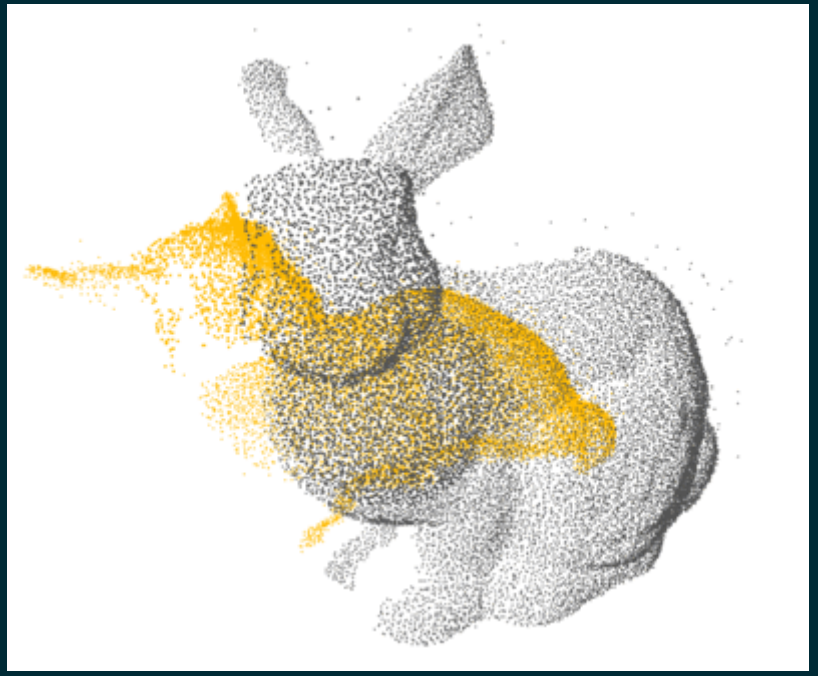
1. POINT REGISTRATION

- Align point clouds P and Q to minimize distance between points (typically L2)
- Either rigid (translation and rotation) or non-rigid transformation as outcome
- Possible issues:
  - noise
  - outliers
  - initial alignment
  - local minima

2. RELATED ALGORITHMS

2.1. IPC

- The classic point registration algorithm
- Finds local minimum based on distance
- What if we steer it towards a global minimum?



2.2. RANSAC

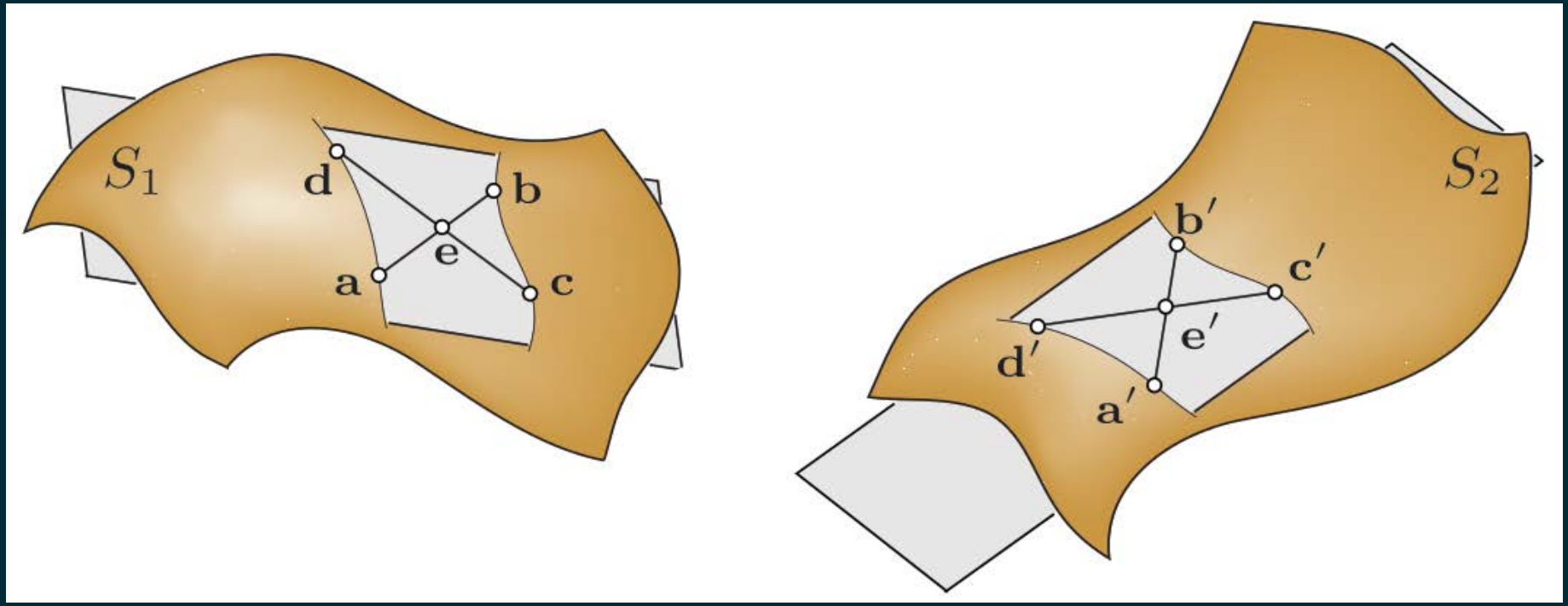
- Use 3-point base, pick in both P and Q simultaneously
- Select base pairs randomly L times, use best fit
- Best fit = max # of points within δ from points in Q

2.3. RANDOMIZED ALIGNMENT

- Finds transforms that align random base to all possible bases in Q
- Tests points in P with this transformation
- Only tests a few points at a time, more if alignment seems good
- As in RANSAC, L picks of random bases

3. 4-POINT CONGRUENT SET REGISTRATION

- Find *wide* 4-point coplanar base in P
- Align with congruent 4-point base in Q



- Only rigid transformation (unless modified)
- Improvement of RANSAC and Randomized Alignment

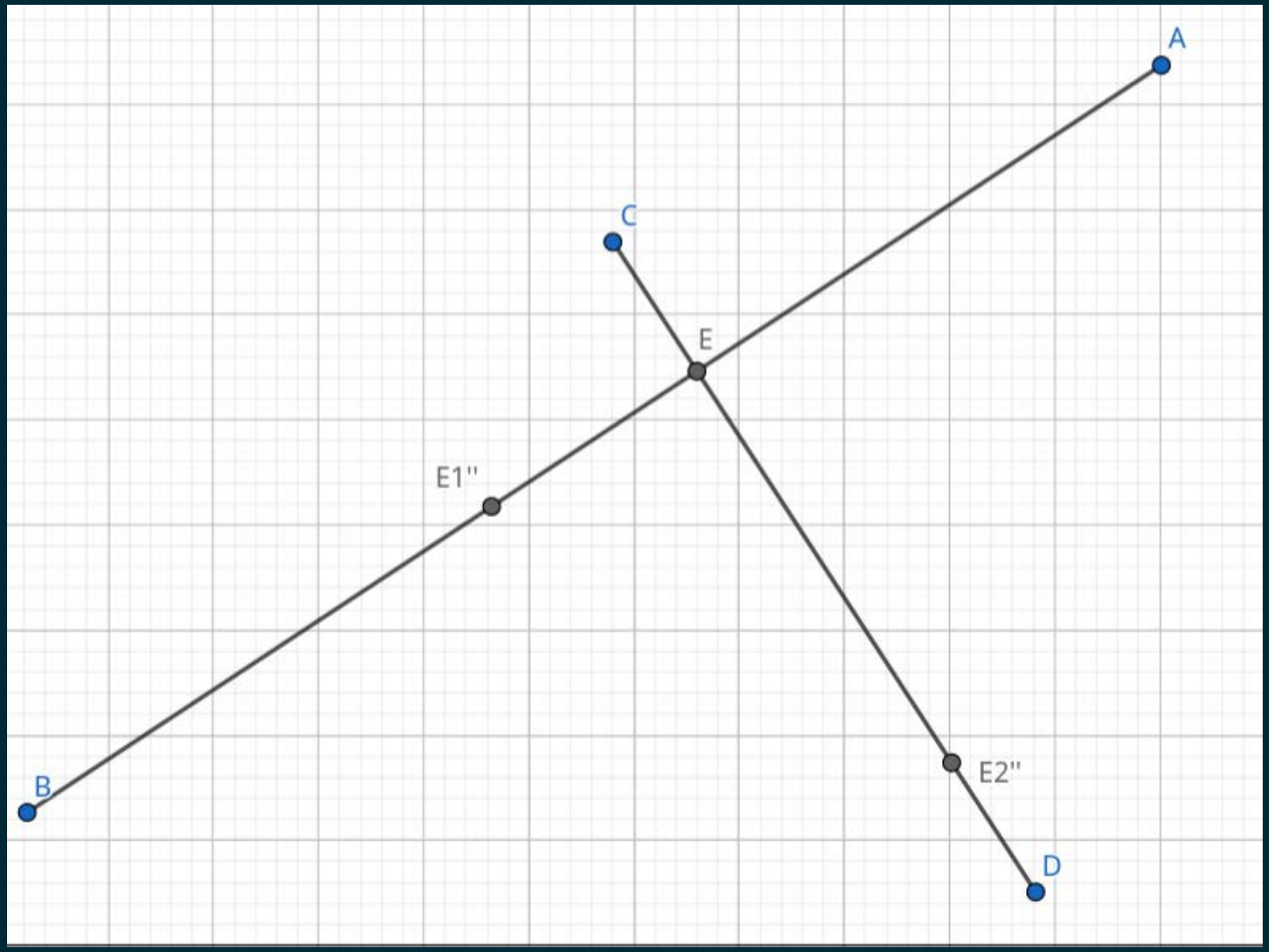
3.1. PARAMETERS

- L random base picks
- Approximation level $\delta > 0$
- Overlap fraction f, determines wideness of base

4. FINDING BASE B IN P

- Allow some non-planarity since perfect planarity is unlikely in real-world data
- Pick 3 random points
    - Alternatively pick 3 points in region of overlap if we know where that region lies

- Select fourth point that forms a *wide* and approximately coplanar base with the rest
  - Need sensible limit of maximum distance between points to ensure base is in overlap
  - Default to decreasing overlap $f = 1., .5, .25, \ldots$


A
C
E
E1''
B
E2''
D

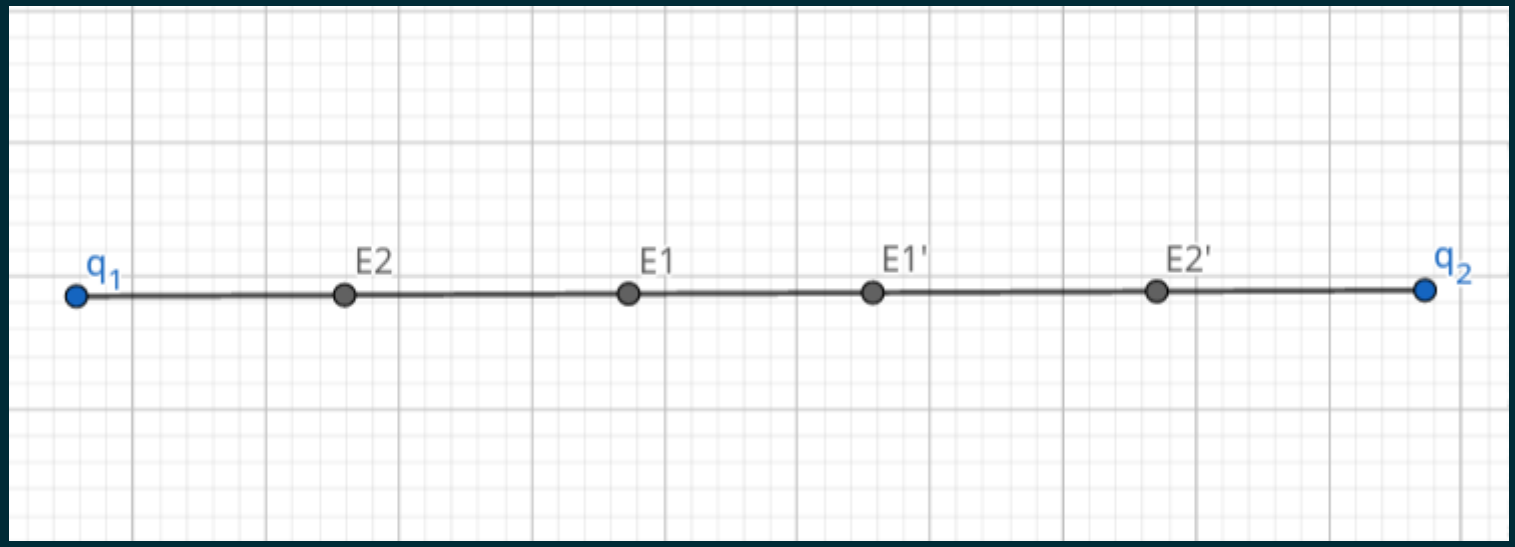
5. EXTRACTING SIMILAR BASES IN Q

5.1. CONCEPT

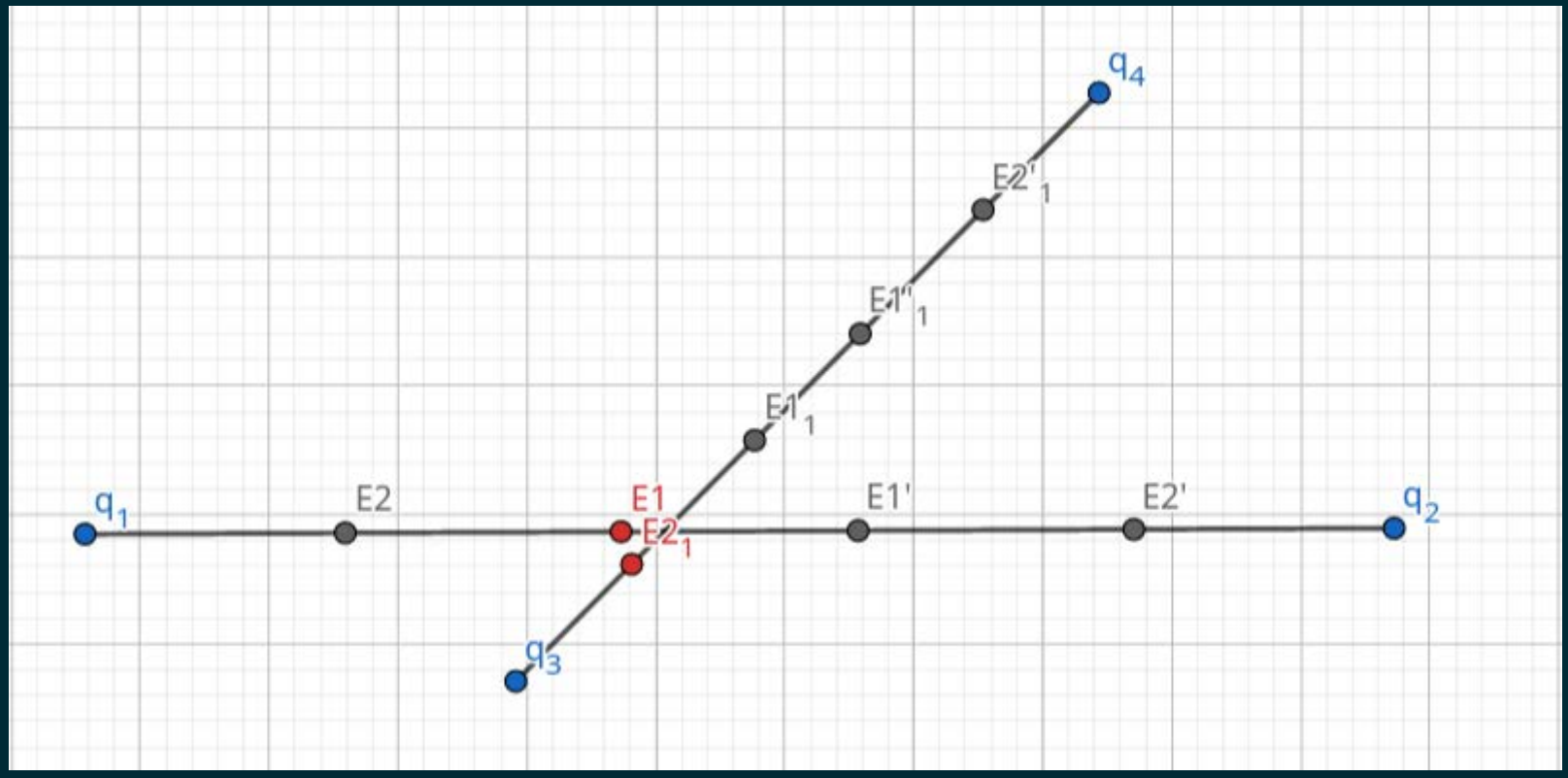
- Two ratios are preserved along the lines between pairs in our 4-point bases, under affine transforms!

$$r_1 = \frac{|\overrightarrow{AE}|}{|\overrightarrow{AB}|} \text{ and } r_2 = \frac{|\overrightarrow{CE}|}{|\overrightarrow{CD}|}$$

- Create intermediate points between points that could constitute base



- When intermediate points are the same ($\pm\delta$), we have a base candidate!



- Whiteboard time!

5.2. IMPLEMENTATION

- Calculate ratios using closest point to intersection between lines in base (since only approx coplanar)
- Find point pairs in Q with same distance ($\pm\delta$) as AB and CD, separately
    - Gives two sets R_1 and R_2

- For all pairs in these two sets, find intermediate points from r_1 and r_2
- Construct range tree for the intermediate points from R_1

- For each intermediate point e stemming from R_2:
  - Use range tree to find points $\leq \delta$ away from the evaluated intermediate point e
  - For all these neighbouring points, add as candidate the 4-point set of the two pairs that e and its neighbour were generated from

6. DETERMINING THE BEST CANDIDATE

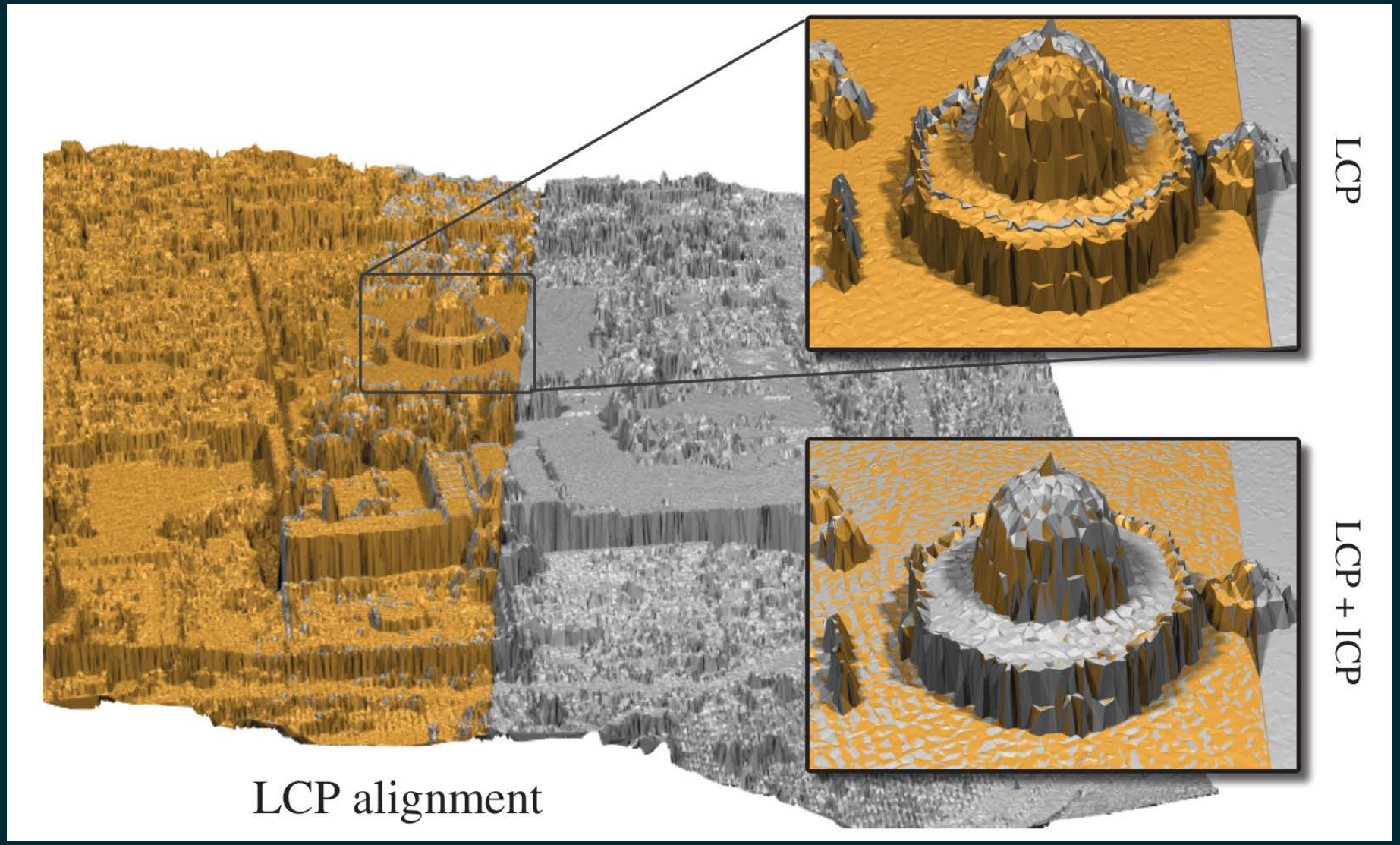
- We now have candidate transformations based on base pair choices.
- For each candidate base U_i:
  - Find transformation T_i that aligns B and U_i
  - Find set S_i of points in P so that distance between Q and the transformed S_i is less than δ

- Best transformation T_i is the one with the maximum number of points within distance δ of Q
  - That is, the transformation with the largest region of overlap!
  - Largest Common Pointset (LCP)
  - Maximize $|S_i|$ and remember corresponding T_i
- Finally, find the best among L attempts with different bases (like RANSAC)

7. REFINEMENTS

7.1. ICP AFTER 4PCS

- ICP can be used to refine alignment, without the cost of doing ICP from scratch

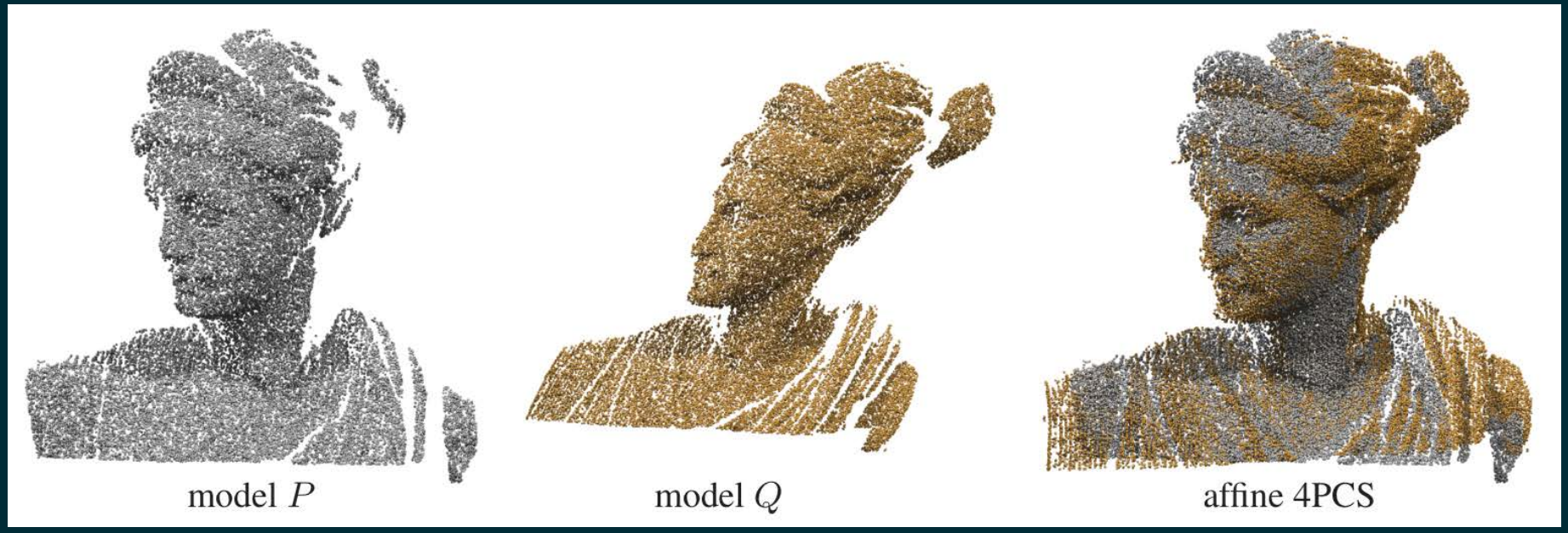

LCP
LCP + ICP
LCP alignment

7.2. USE SURFACE NORMALS

- Limit R_1 and R_2 to point pairs with normals that have angles $\approx \angle(N_A, N_B)$ or $\angle(N_C, N_D)$
- Possibly 2x speedup

7.3. SUPPORTING AFFINE TRANSFORMS

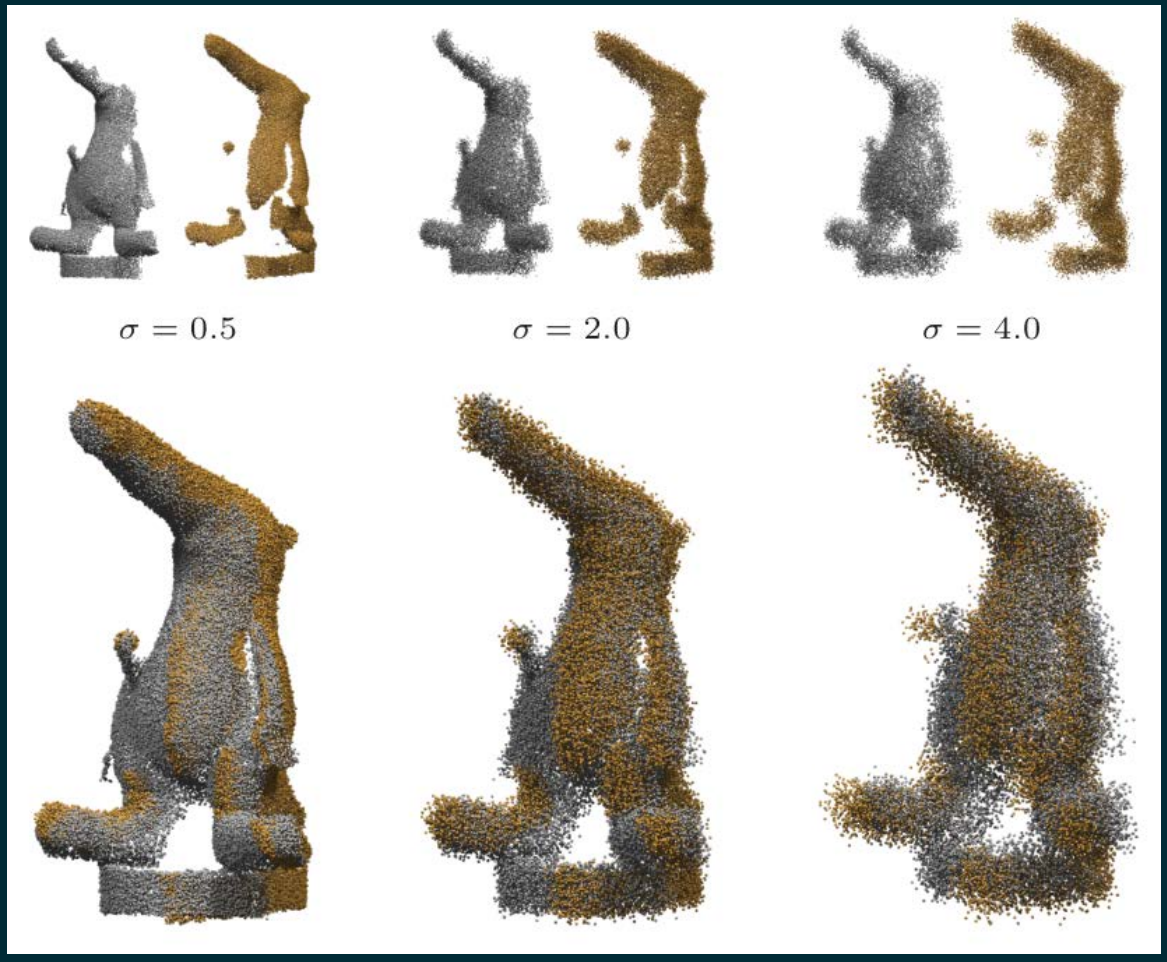
- For undoing *small* shears
- Use two bases with one common point
- Possibly $O(n^2)$ space complexity for general affine transforms :(



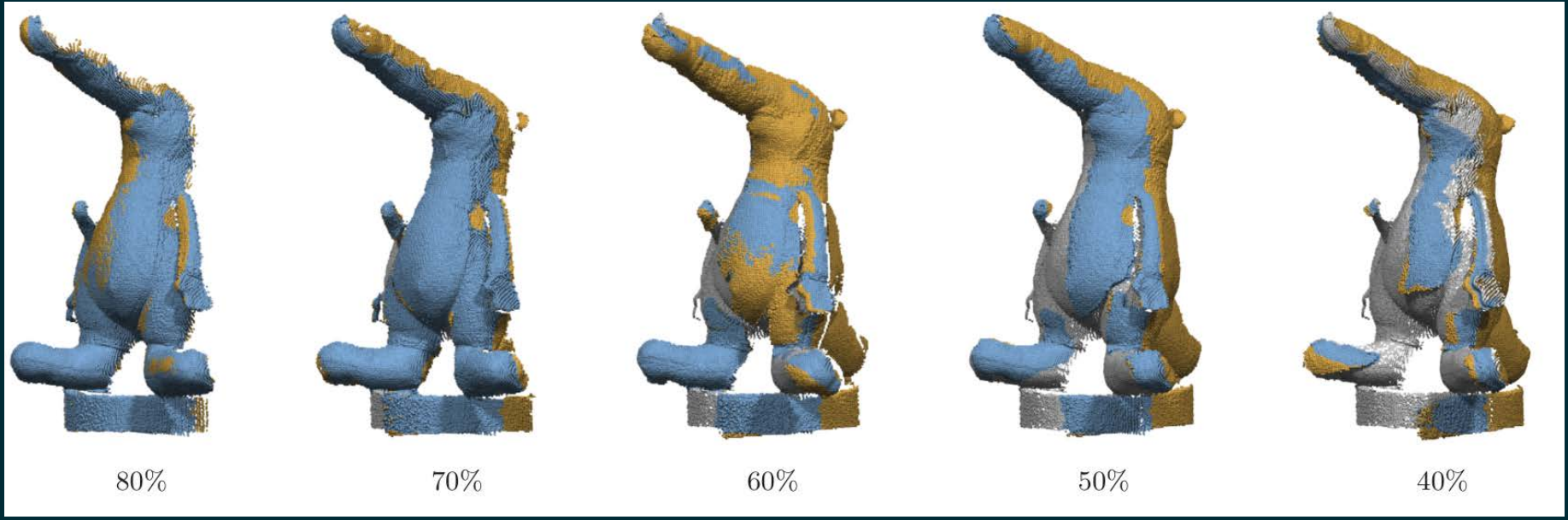
8. RESULTS

- An algorithm with
  - time complexity $O(n^2)$ (improvement over others)
  - space complexity $O(n)$
  - tolerance for noise
  - tolerance for low overlap

- No need to remove noise before 4PCS

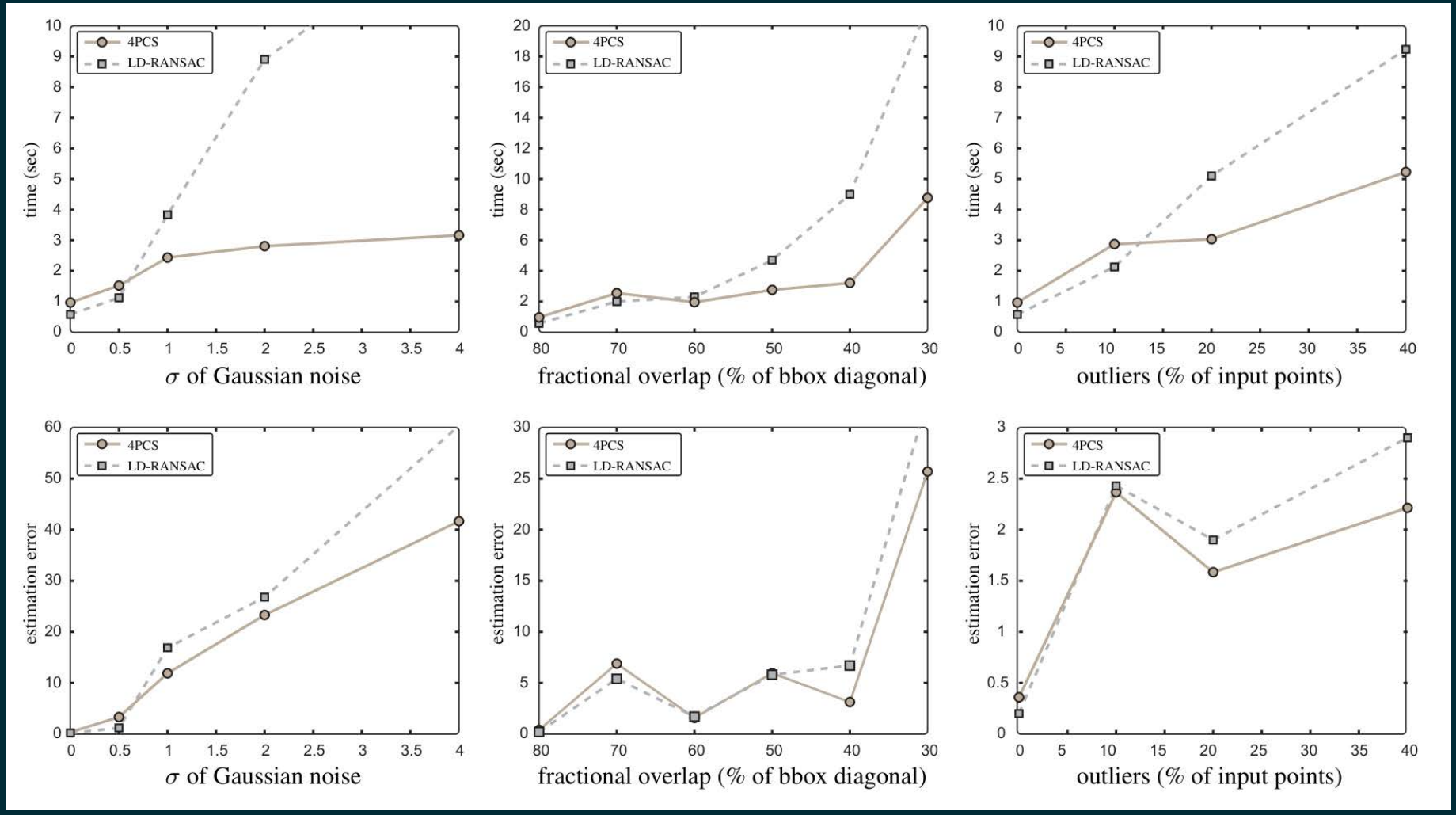


- Degrades slightly with lower overlap
- Subsequent ICP still results in good alignment



- Performs well with outliers

- Outperforms LD-RANSAC in cases with
  - noisy data
  - low overlap
  - lots of outliers


4PCS
LD-RANSAC
time (sec)
σ of Gaussian noise
fractional overlap (% of bbox diagonal)
outliers (% of input points)
estimation error