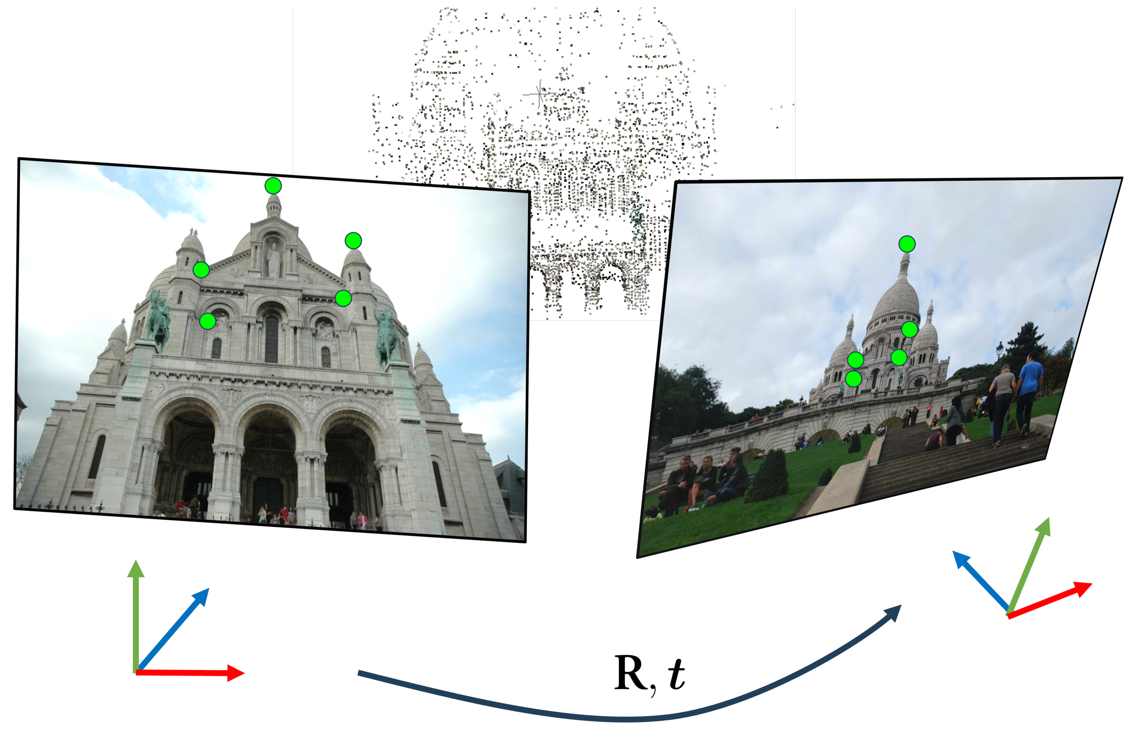
# Fast Relative Pose Estimation using Relative Depth


JONATHAN ASTERMARK[1], YAQING DING[1,2], VIKTOR LARSSON[1], AND ANDERS HEYDEN[1]

[1]CVML, Centre for Mathematical Sciences, Lund University
[2]Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague


## Summary


**Goal:** Improve estimation of the relative camera pose $(\mathbf{R}, \boldsymbol{t})$ between two images.




- Given a sparse set of keypoint correspondences, the relative camera pose can be estimated using RANSAC.
- For each point-correspondence, in addition to the positions $(x, y)$, we use the *relative depth*, i.e. relative distance to the same scene point in the two images.
- Using this extra constraint we can generate pose candidates for RANSAC using fewer point correspondences, compared to purely coordinate-based solvers.

### Contributions

- A novel 3-point minimal solver for relative pose, using relative depths.
- We show that the relative depth can either be estimated from SIFT scales, or predicted using a simple neural network.
- Through experiments, we demonstrate that the smaller sample size leads to a significantly reduced runtime in settings with high outlier ratios, compared to purely point-based solvers.


## Relative Pose Estimation

The projections $\boldsymbol{x}, \boldsymbol{x}'$ of a 3D-point $\boldsymbol{X}$ are described by the camera equations

$$\begin{cases} \lambda \boldsymbol{x} = \boldsymbol{X} \\ \lambda' \boldsymbol{x}' = \mathbf{R}\boldsymbol{X} + \boldsymbol{t} \end{cases} \Rightarrow \quad \lambda' \boldsymbol{x}' = \lambda \mathbf{R} \boldsymbol{x} + \boldsymbol{t}, \qquad (1)$$

where $\lambda$ and $\lambda'$ are the depths of point $\boldsymbol{X}$.

- Classical minimal solver **requires 5 points** to estimate relative pose.
- In RANSAC, number of iterations grows exponentially with sample size.

## Relative Depth in Relative Pose Estimation

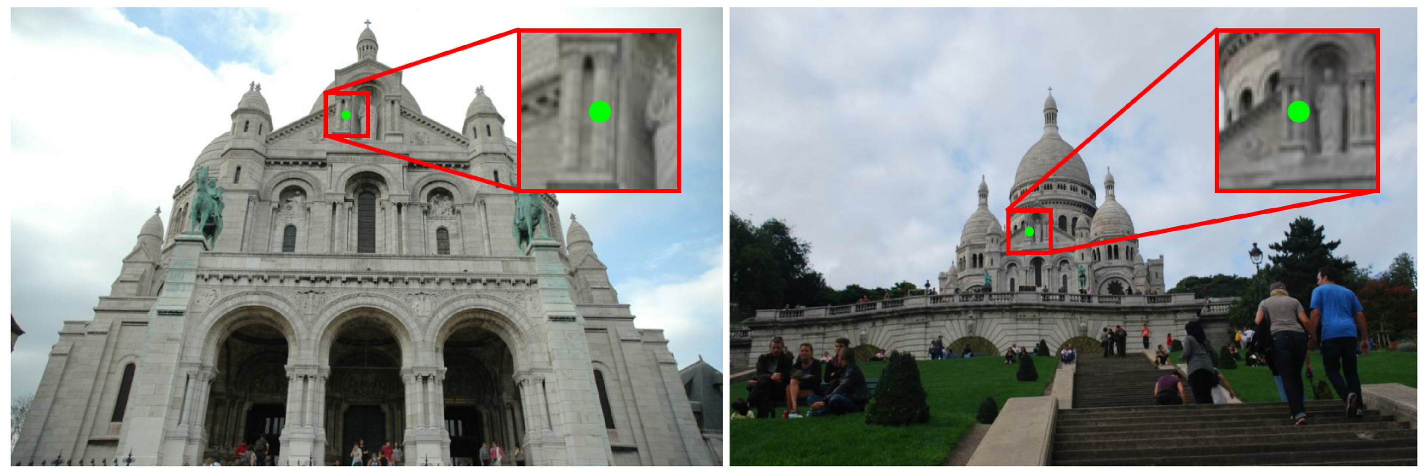
**Idea:** Leverage relative depth constraints, observed from scale changes.

- If we introduce relative depth $\sigma := \lambda'/\lambda$, we can rewrite (1) as

$$\lambda(\sigma \boldsymbol{x}' - \mathbf{R}\boldsymbol{x}) = \boldsymbol{t}. \qquad (2)$$

- Relative depth inversely proportional to the relative scale in the images

$$\sigma := \frac{\lambda'}{\lambda} = \frac{f'}{f}\frac{s}{s'}. \qquad (3)$$



- Keypoint detection scale (e.g. from SIFT) can be used directly in (3).
- From (2), we introduce a novel minimal 3-point solver

$$\begin{aligned} \sigma_1 \lambda_1 \boldsymbol{x}_1' &= \lambda_1 \mathbf{R} \boldsymbol{x}_1 + \boldsymbol{t}, \\ \sigma_2 \lambda_2 \boldsymbol{x}_2' &= \lambda_2 \mathbf{R} \boldsymbol{x}_2 + \boldsymbol{t}, \\ \lambda_3' \boldsymbol{x}_3' &= \lambda_3 \mathbf{R} \boldsymbol{x}_3 + \boldsymbol{t}. \end{aligned}$$

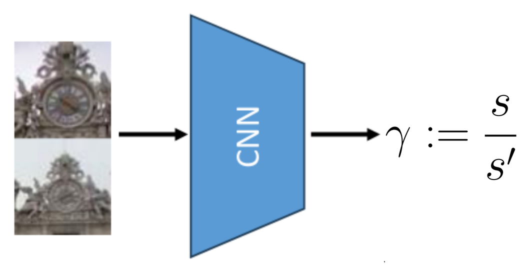
- Forming the differences and taking the norm eliminates $\mathbf{R}$ and yields

$$\begin{aligned} \|\sigma_1 \boldsymbol{x}_1' - \sigma_2 \lambda_2 \boldsymbol{x}_2'\|^2 &= \|\boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2\|^2, \\ \|\sigma_1 \boldsymbol{x}_1' - \lambda_3' \boldsymbol{x}_3'\|^2 &= \|\boldsymbol{x}_1 - \lambda_3 \boldsymbol{x}_3\|^2, \\ \|\sigma_2 \lambda_2 \boldsymbol{x}_2' - \lambda_3' \boldsymbol{x}_3'\|^2 &= \|\lambda_2 \boldsymbol{x}_2 - \lambda_3 \boldsymbol{x}_3\|^2. \end{aligned}$$

- We can find the three unknowns (red) by solving two quadratics.
- In the paper we also show extension to known vertical direction (2-points).

## RelScaleNet

To improve scale estimate, we introduce a simple neural network that directly regresses relative scale from pairs of image patches.

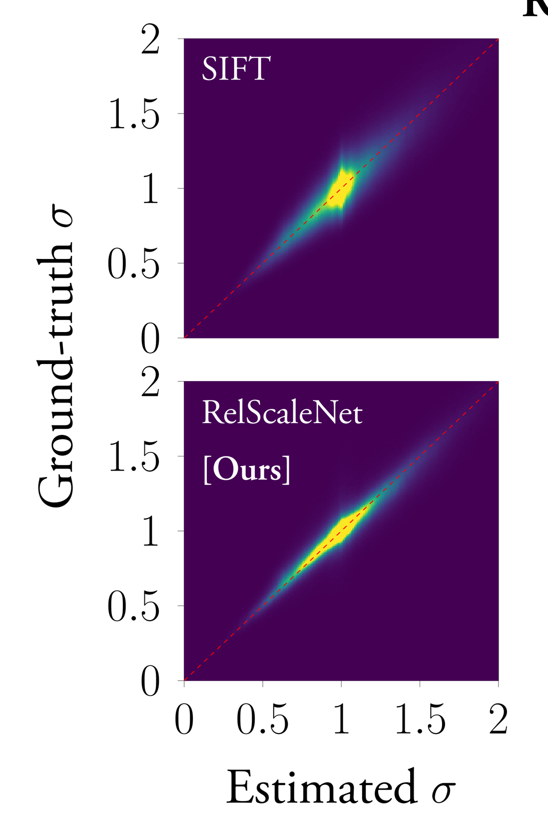



**Input:** A pair of image patches from neighborhood of corresponding points.

**Output:** Estimate of relative scale $\gamma = s/s'$.

- We train on MegaDepth and supervise with MSE-loss w.r.t. ground-truth $\gamma$.
- Ground-truth relative depth calculated from Structure-from-Motion model.

## Evaluation




**Relative Depth Estimation**

| | | Method | Med.↓ | Accuracy @0.05↑ | Accuracy @0.1↑ | Accuracy @0.2↑ |
|---|---|---|---|---|---|---|
| IMC-PT | SIFT | SIFT | 0.063 | 0.41 | 0.69 | 0.92 |
| | | Self-Sca-Ori | 0.274 | 0.12 | 0.22 | 0.40 |
| | | RelScaleNet [OURS] | **0.033** | **0.65** | **0.86** | **0.97** |
| ScanNet-1500 | SIFT | SIFT | 0.071 | 0.38 | 0.64 | 0.88 |
| | | Self-Sca-Ori | 0.120 | 0.27 | 0.45 | 0.66 |
| | | RelScaleNet [OURS] | **0.044** | **0.55** | **0.80** | **0.94** |
| | SP+SG | Self-Sca-Ori | 0.201 | 0.19 | 0.32 | 0.50 |
| | | RelScaleNet [OURS] | **0.114** | **0.27** | **0.46** | **0.68** |

**Relative Pose Estimation with LO/GC-RANSAC on IMC-PT**

| RSC | Method | All pairs AUC@5° | All pairs RT(ms) | Hardest 5% AUC@5° | Hardest 5% RT(ms) |
|---|---|---|---|---|---|
| LO-RANSAC | 5 pt. (Nistér) | **56.89** | 15.7 | **12.13** | 42.2 |
| | 3 pt. + SIFT (Barath & Kukelova) | 30.77 | **7.0** | 1.23 | 21.3 |
| | 3 pt. + SIFT [OURS] | 54.30 | 13.4 | 8.72 | **2.8** |
| | 3 pt. + RelScaleNet [OURS] | 54.63 | 15.0 | 9.47 | **2.8** |
| GC-RANSAC | 5 pt. (Nistér) | **56.22** | 25.4 | **9.76** | 16.5 |
| | 3 pt. + SIFT (Barath & Kukelova) | 50.55 | **11.1** | 2.16 | 6.0 |
| | 3 pt. + SIFT [OURS] | 52.73 | 16.2 | 5.24 | **4.7** |
| | 3 pt. + RelScaleNet [OURS] | 53.11 | 16.8 | 5.65 | **4.7** |

**Relative Pose Estimation with LO-RANSAC on ScanNet-1500**

| KP. | Method | AUC@5° | AUC@10° | AUC@20° | Runtime (ms) |
|---|---|---|---|---|---|
| SIFT | 5 pt. (Nistér) | **11.06** | **21.99** | **33.32** | 8.2 |
| | 3 pt. + SIFT (Barath & Kukelova) | 4.94 | 10.33 | 17.16 | 3.7 |
| | 3 pt. + SIFT [OURS] | 9.90 | 20.59 | 31.96 | **2.9** |
| | 3 pt. + RelScaleNet [OURS] | 10.43 | 21.21 | 32.43 | **2.9** |
| SP+SG | 5 pt. (Nistér) | 17.55 | 34.21 | 51.50 | 59.4 |
| | 3 pt. + RelScaleNet [OURS] | **18.39** | **35.46** | **52.24** | **10.4** |

### Conclusion

Our novel 3-point solver has similar accuracy to the 5-point solver, while being significantly faster in high outlier settings.