Triangulation: Why Optimize?

Seong Hun Lee and Javier Civera
I3A, University of Zaragoza, Spain

Universidad Zaragoza

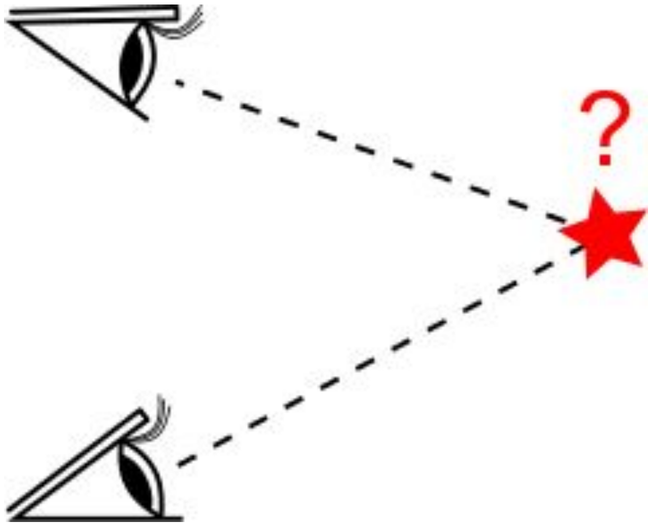
1542

unizar

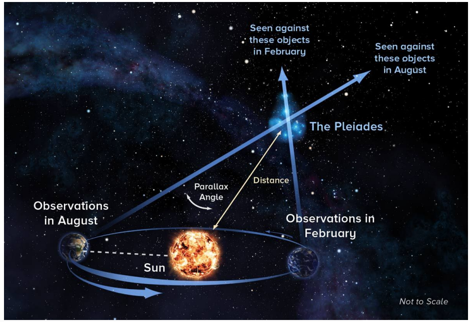
Triangulation:

Locating a 3D point observed from a known baseline.

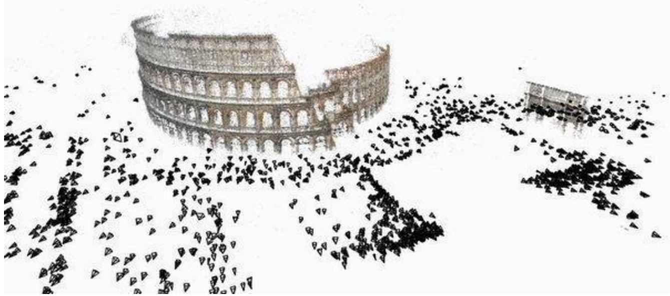
Used in surveying, astrometry, navigation, computer vision, etc.



Sorn340, shutterstock.com

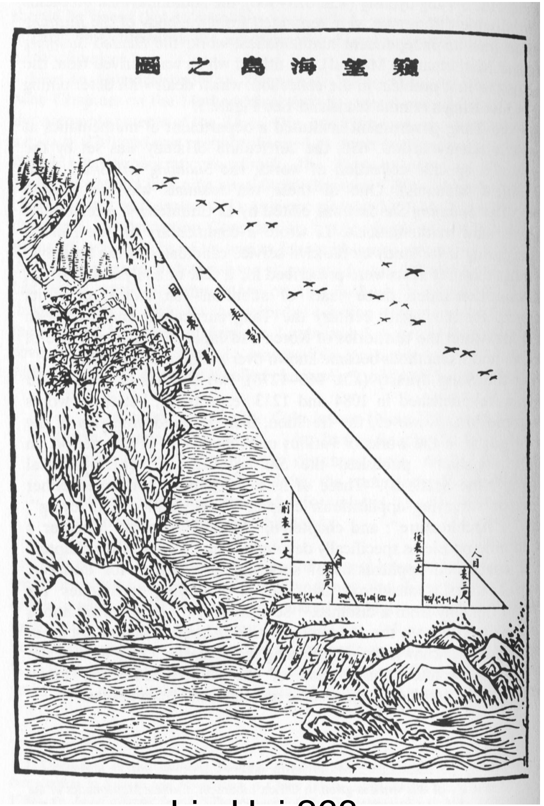


Bill Saxton, NRAO/AUI/NSF

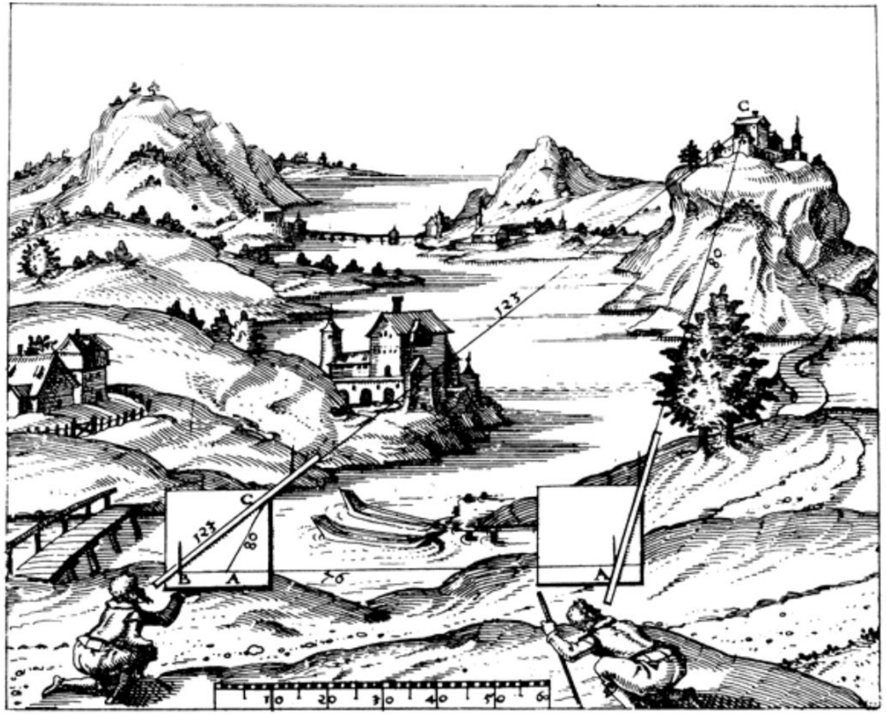


Building Rome in a day.
https://grail.cs.washington.edu/rome/

Triangulation is much older than computer vision…!



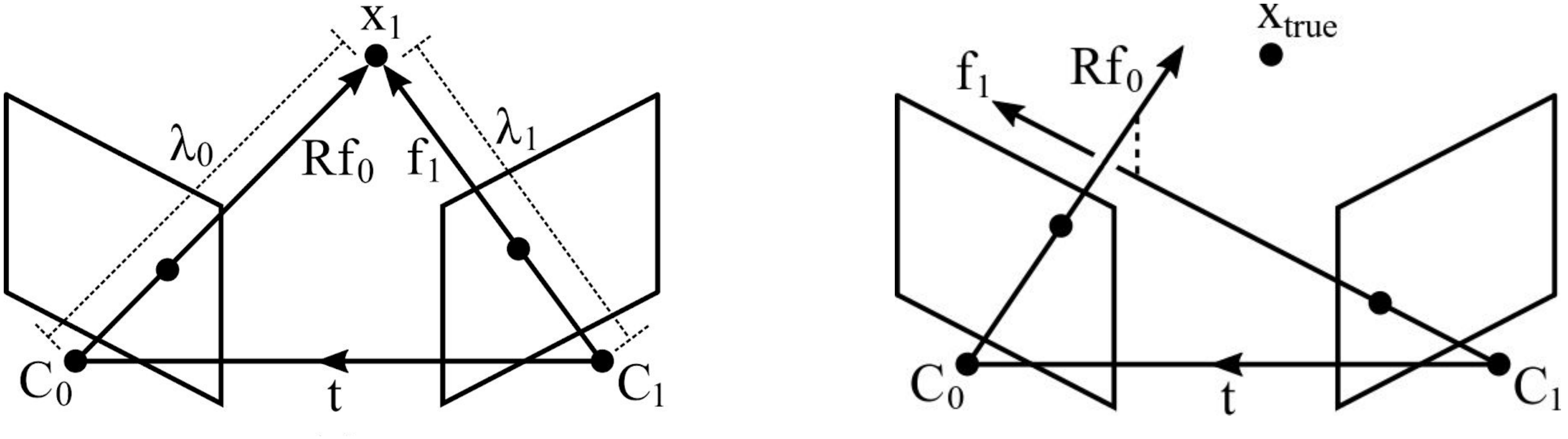
Liu Hui 263.
(Illustrated in 1726)



Zubler 1625

Two-View Triangulation

Locating the 3D point given its projections
in two views of known calibration and pose.

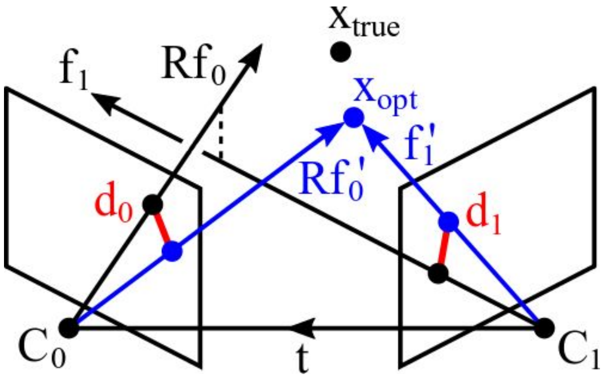


(a) Ideal (b) Real-world

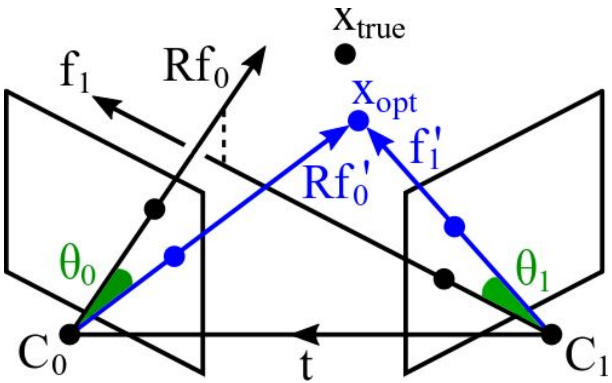
Two-View Triangulation

1. Optimal methods:

Find intersecting rays that minimize the reprojection cost.



(a) Image Reprojection error



(b) Angular Reprojection error

- L_1 norm: $d_0 + d_1$
- L_2 norm: ${d_0}^2 + {d_1}^2$
- L_∞ norm: $\max(d_0 + d_1)$

[Hartley 1997, Nister 2001]

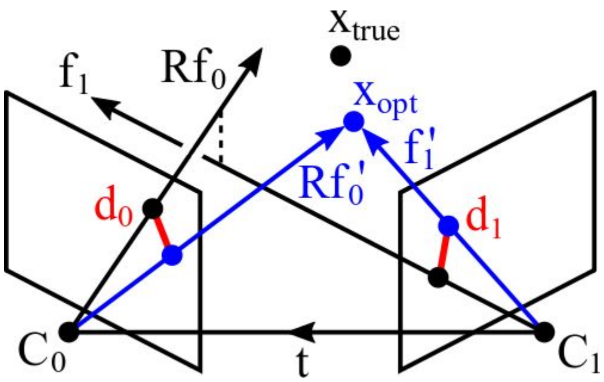
- L_1 norm: $\theta_0 + \theta_1$
- L_2 norm: $\sin^2(\theta_0) + \sin^2(\theta_1)$
- L_∞ norm: $\max(d_0 + d_1)$

[Oliensis 2002, Lee 2019]

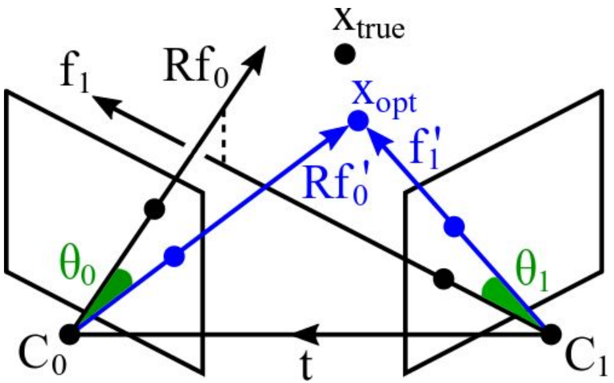
Two-View Triangulation

1. Optimal methods:

Find intersecting rays that minimize the reprojection cost.



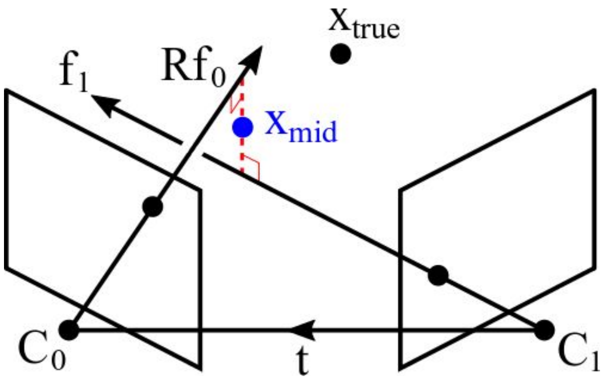
(a) Image Reprojection error



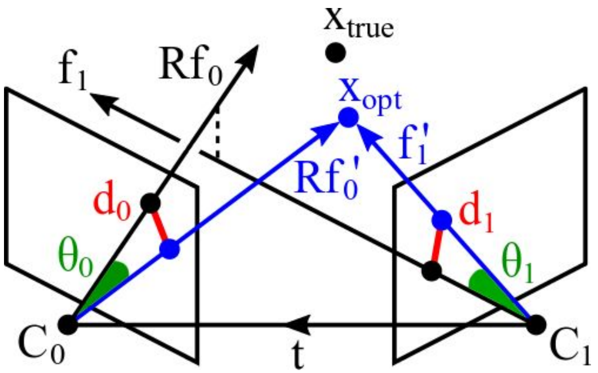
(b) Angular Reprojection error

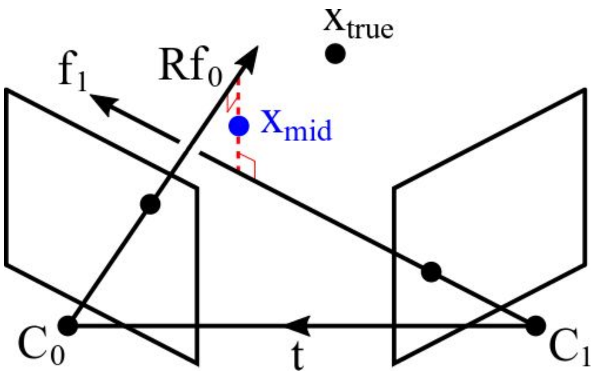
2. Midpoint method:

Find the midpoint of the common perpendicular.



Two-View Triangulation

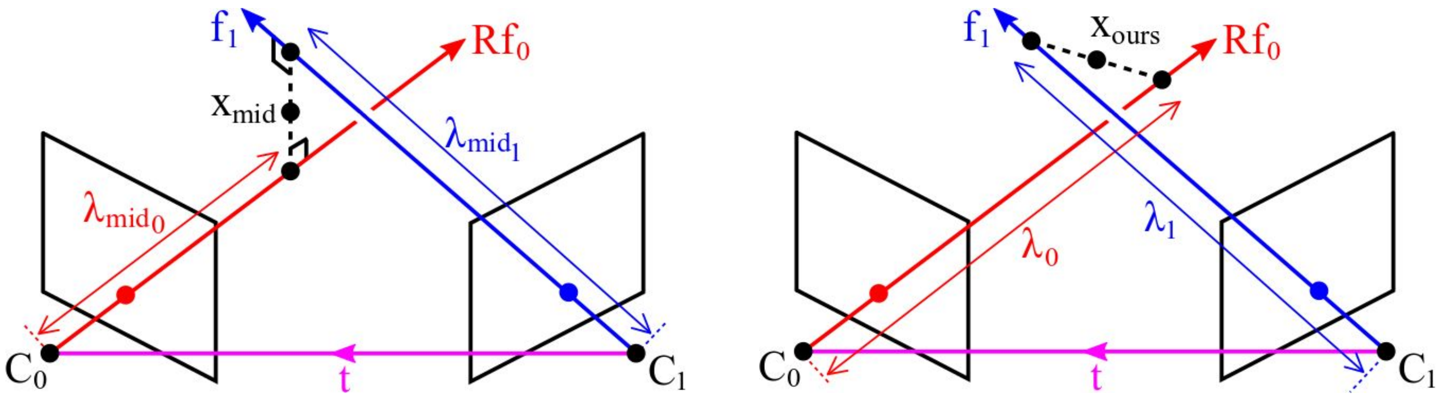




Optimal methods	**Midpoint method**
✔ Fast [Lindstrom 10, Lee 19] ✔ Minimal 2D (or angular) reproj. error ✔ Small 3D error at high parallax ✘ Large 3D error at low parallax	✔ Faster ✘ Large 2D error at low parallax ✔ Small 3D error at high parallax ✔ Smaller 3D error (than the optimal method) at low parallax

The discrepancy between 2D and 3D accuracy was reported in [Hartley 97]...!

Contribution #1: Alternative Midpoint Method



Let $\mathbf{p} = \mathbf{R}\widehat{\mathbf{f}}_0 \times \widehat{\mathbf{f}}_1$, $\mathbf{q} = \mathbf{R}\widehat{\mathbf{f}}_0 \times \mathbf{t}$ and $\mathbf{r} = \widehat{\mathbf{f}}_1 \times \mathbf{t}$.

Classic Midpoint Depths

$$\lambda_{\text{mid}0} = \frac{\widehat{\mathbf{p}} \cdot \mathbf{r}}{\|\mathbf{p}\|}, \quad \lambda_{\text{mid}1} = \frac{\widehat{\mathbf{p}} \cdot \mathbf{q}}{\|\mathbf{p}\|}$$

Proposed Depths

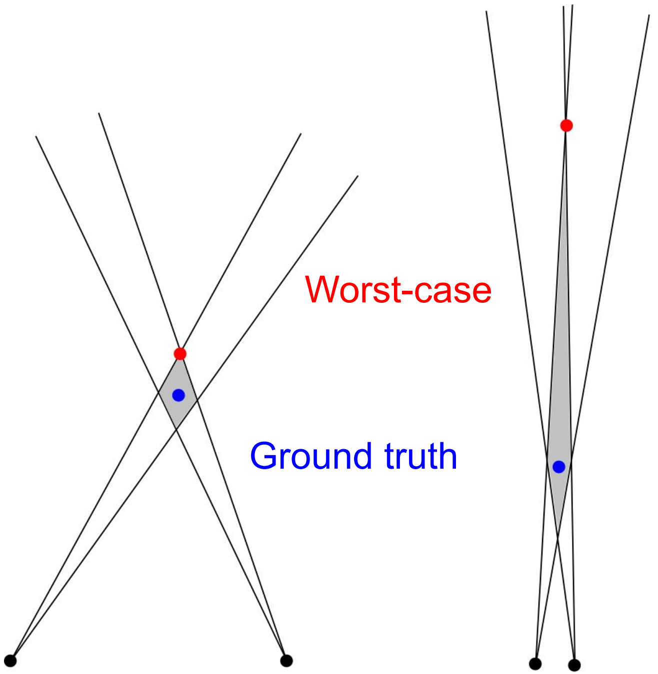
$$\lambda_0 = \frac{\|\mathbf{r}\|}{\|\mathbf{p}\|}, \quad \lambda_1 = \frac{\|\mathbf{q}\|}{\|\mathbf{p}\|}$$

- When the rays intersect, this is just the sine rule.
- $\lambda_{\text{mid}0} \leq \lambda_0$ and $\lambda_{\text{mid}1} \leq \lambda_1$

Claim: Our method is better at low parallax!

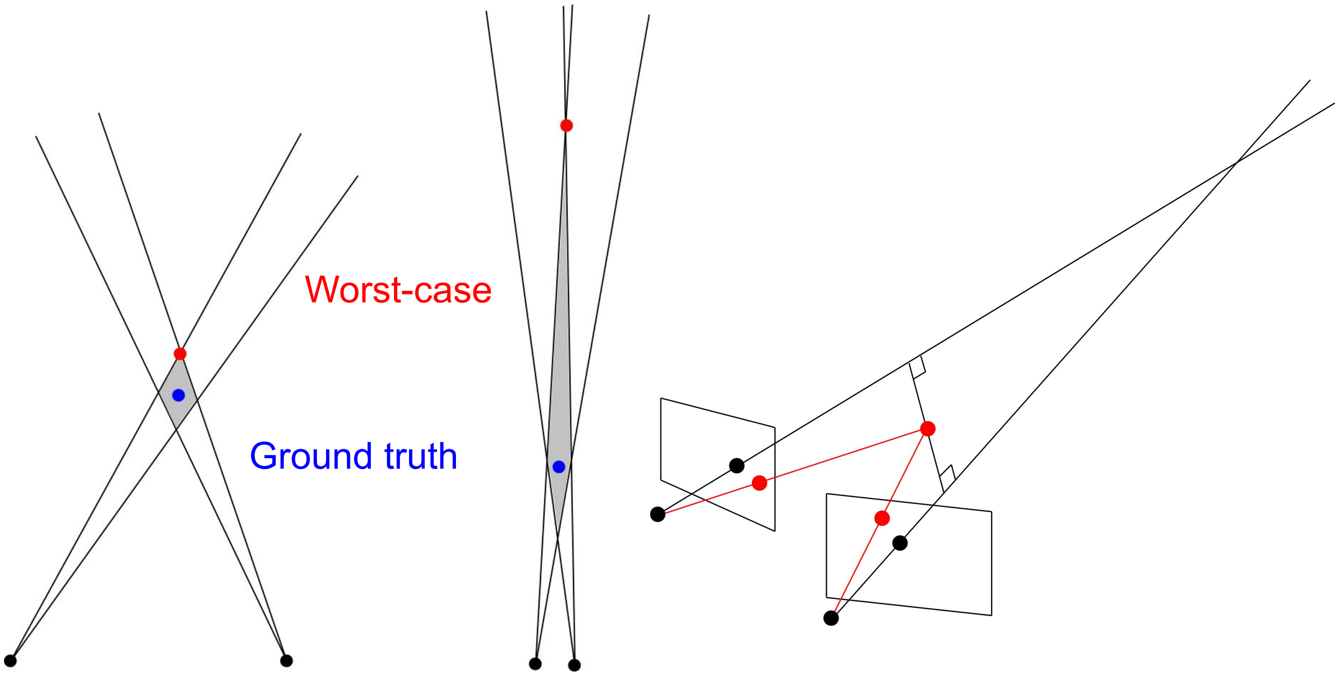
Claim: Our method is better at low parallax!

- Optimal methods tend to locate the point too far → **Large 3D error!**



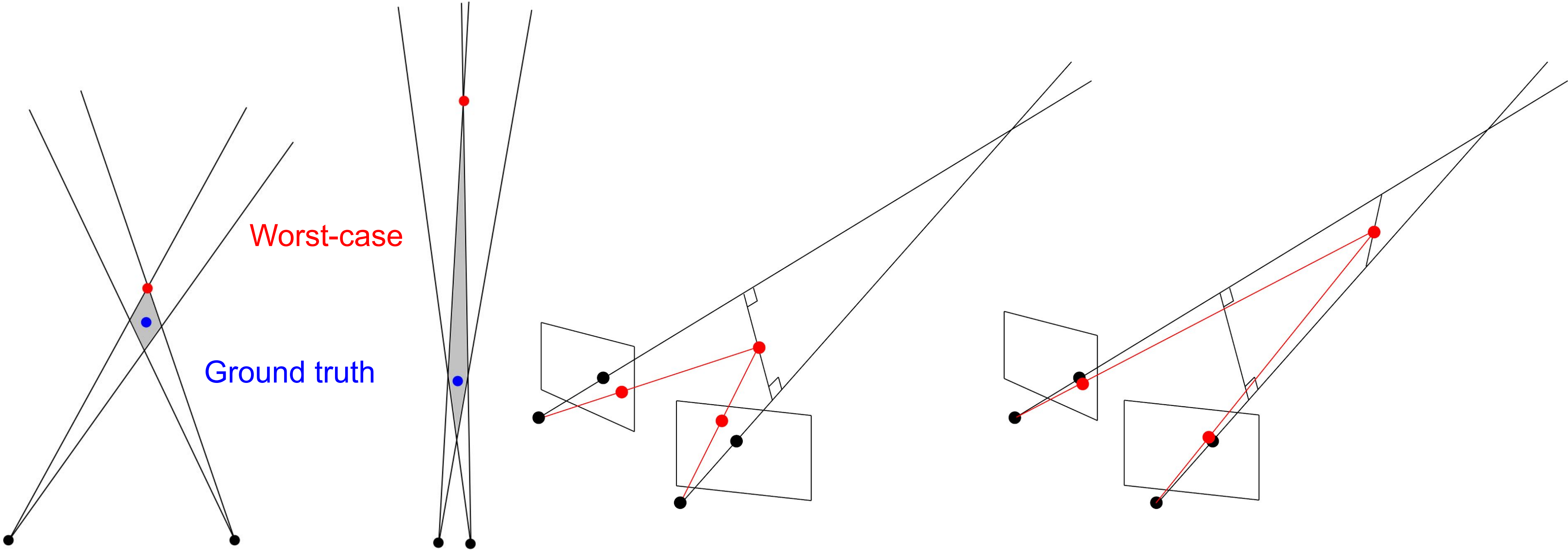
Claim: Our method is better at low parallax!

- Optimal methods tend to locate the point too far → **Large 3D error!**
- Midpoint method tends to locate the point too close → **Large 2D error!**



Claim: Our method is better at low parallax!

- Optimal methods tend to locate the point too far → **Large 3D error!**
- Midpoint method tends to locate the point too close → **Large 2D error!**
- Recall $\lambda_{\text{mid}0} \leq \lambda_0$ and $\lambda_{\text{mid}1} \leq \lambda_1$ → Our midpoint is farther, but not too far.
  → **Small 2D and 3D error (sweet spot)!**



Contribution #2: Alternative Cheirality Check

Classic Midpoint Depths

$$\lambda_{\text{mid}0} = \frac{\widehat{\mathbf{p}} \cdot \mathbf{r}}{\|\mathbf{p}\|}, \quad \lambda_{\text{mid}1} = \frac{\widehat{\mathbf{p}} \cdot \mathbf{q}}{\|\mathbf{p}\|}$$

Inaccurate measurements can lead to negative depths. When this happens, simply discard the point.

Proposed Depths

$$\lambda_0 = \frac{\|\mathbf{r}\|}{\|\mathbf{p}\|}, \quad \lambda_1 = \frac{\|\mathbf{q}\|}{\|\mathbf{p}\|}$$

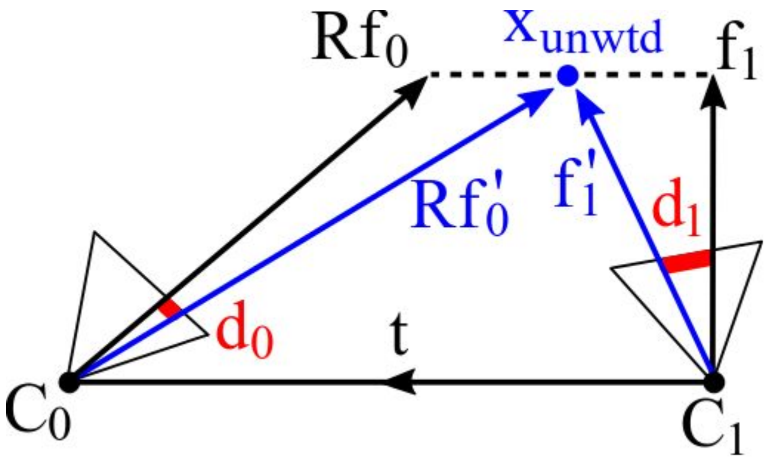
The depths are always positive.

Discard the point if assuming a negative depth brings the two points on each ray closer together, i.e.,

$$\|\mathbf{t}+\lambda_0\mathbf{R}\widehat{\mathbf{f}}_0-\lambda_1\widehat{\mathbf{f}}_1\|^2 \geq \min\left(\|\mathbf{t}+\lambda_0\mathbf{R}\widehat{\mathbf{f}}_0+\lambda_1\widehat{\mathbf{f}}_1\|^2, \|\mathbf{t}-\lambda_0\mathbf{R}\widehat{\mathbf{f}}_0-\lambda_1\widehat{\mathbf{f}}_1\|^2, \|\mathbf{t}-\lambda_0\mathbf{R}\widehat{\mathbf{f}}_0+\lambda_1\widehat{\mathbf{f}}_1\|^2\right)$$

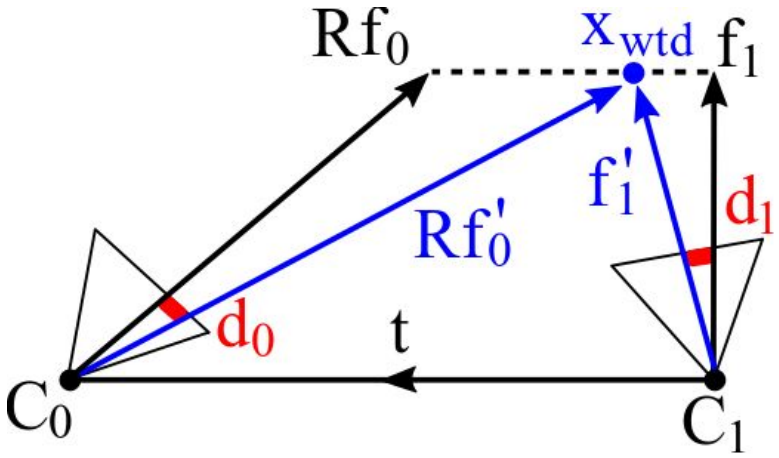
Contribution #3: Inverse Depth Weighting

Unweighted Midpoint



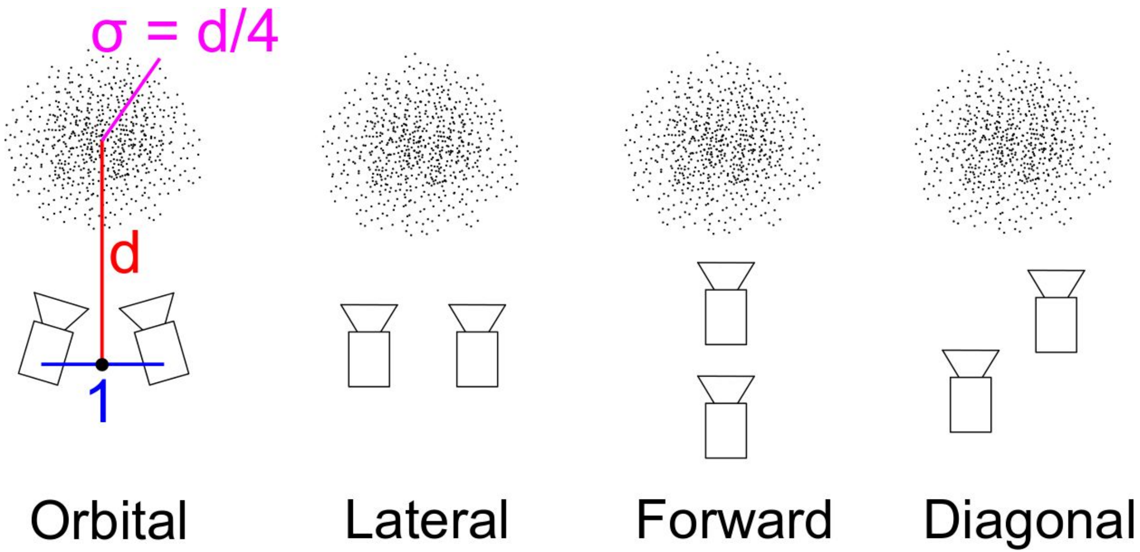
$$x_{\text{unwtd}} = \frac{\mathbf{t} + \lambda_0 \mathbf{R}\widehat{\mathbf{f}_0} + \lambda_1 \widehat{\mathbf{f}_1}}{2}$$

Inverse Depth Weighted Midpoint



$$x_{\text{wtd}} = \frac{{\lambda_0}^{-1}(\mathbf{t} + \lambda_0 \mathbf{R}\widehat{\mathbf{f}_0}) + {\lambda_1}^{-1}(\lambda_1 \widehat{\mathbf{f}_1})}{{\lambda_0}^{-1} + {\lambda_1}^{-1}}$$

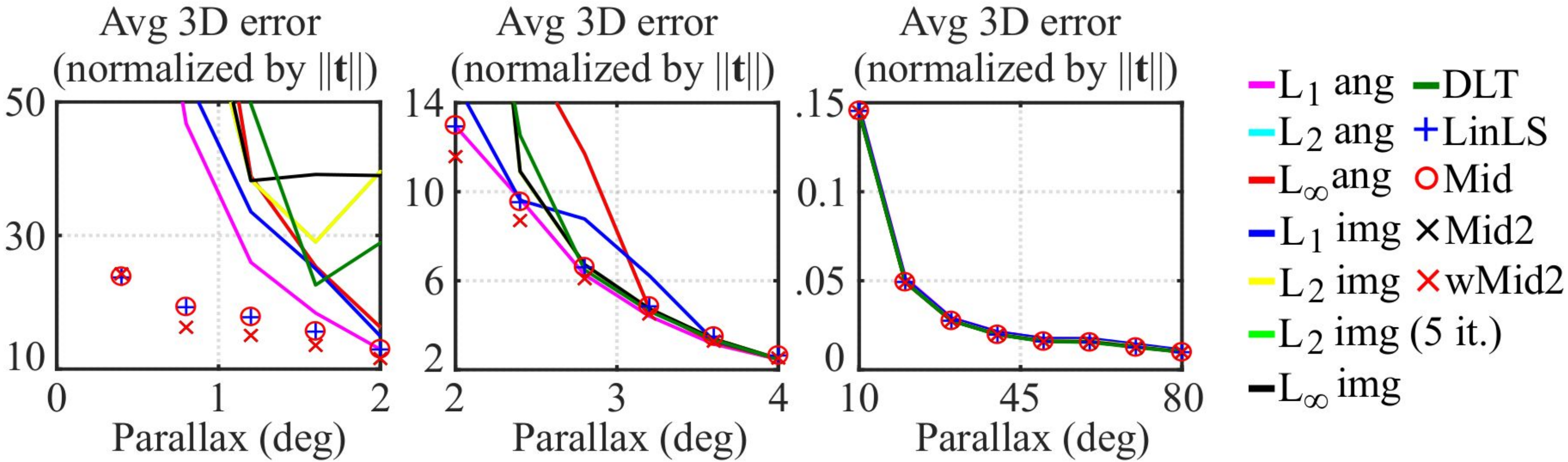
Synthetic Dataset



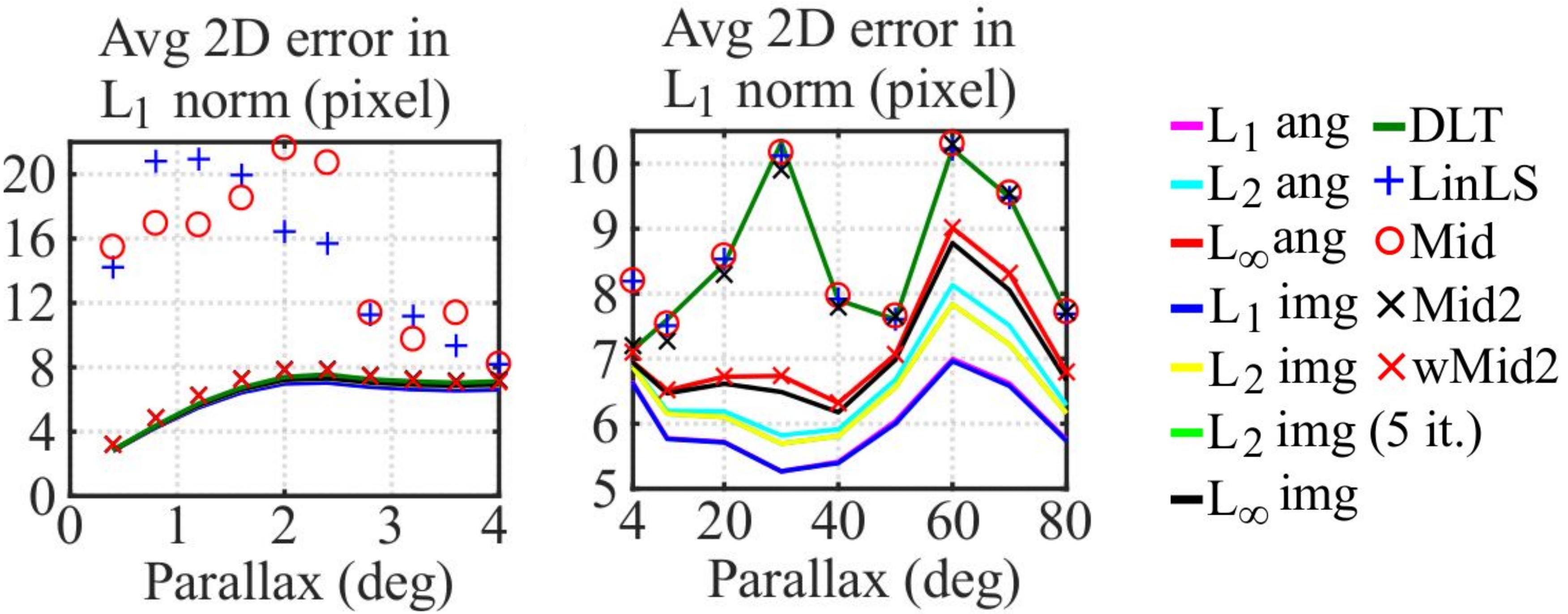
- d = {0.5, 1, 2, …, 64} x baseline
- Image noise level = 1, 2, 3, …, 8 pix.

Over a million unique triangulation problems

3D error evaluation on synthetic dataset



2D error evaluation on synthetic dataset



Speed

Method	Speed (Points/ sec)
Classic Midpoint	38M
L1 angular optimal [Lee 19]	29M
Ours (unweighted)	**21M**
L∞ angular angular [Lee 19]	13M
Ours (weighted)	**12M**
L2 image optimal [Lindstrom 10]	550K

Summary

Optimal Methods	Classic Midpoint	Our Weighted Midpoint
✔ Fast [Lindstrom 10, Lee 19] ✔ Minimal 2D error ✔ Small 3D error at high parallax ✘ Large 3D error at low parallax	✔ Fastest ✘ Large 2D error ✔ Small 3D error at high parallax ✔ Smaller 3D error (than the optimal methods) at low parallax	✔ Fast ✔ Small 2D error ✔ Small 3D error at high parallax ✔ Smaller 3D error (than the optimal methods) at low parallax

Why Optimize?

- For parallax < 4 deg, DO NOT OPTIMIZE and use our unweighted midpoint method instead.
    - Although our method is not optimal in a geometrically meaningful way, it clearly outperforms the existing optimal and non-optimal methods at low parallax.
    - Inverse depth weighting does not help much at low parallax.
- For parallax > 4 deg, DO OPTIMIZE using L1 angular method [Lee 19].
    - Similar 3D accuracy, yet optimal in L1 angular reproj error and as fast as the midpoint.

Thank you!