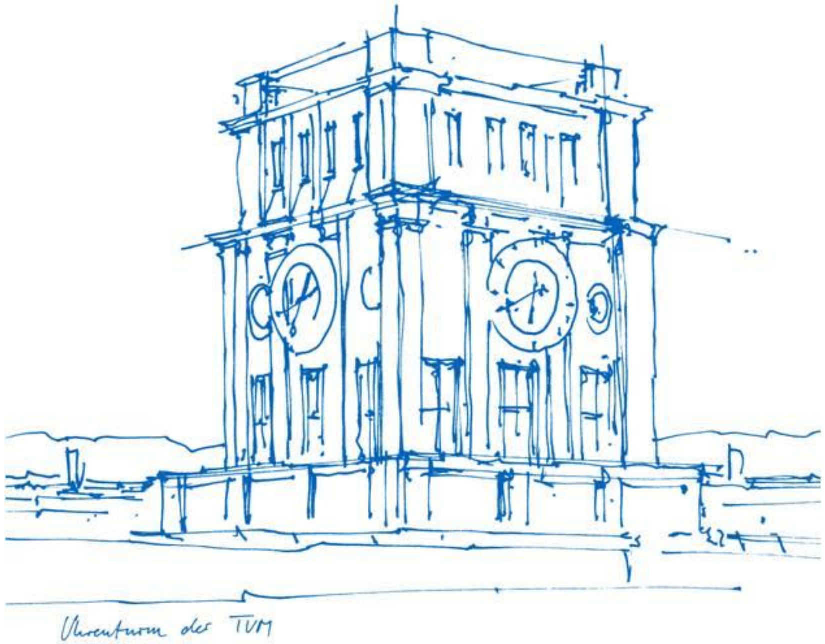
Direct Sparse Odometry

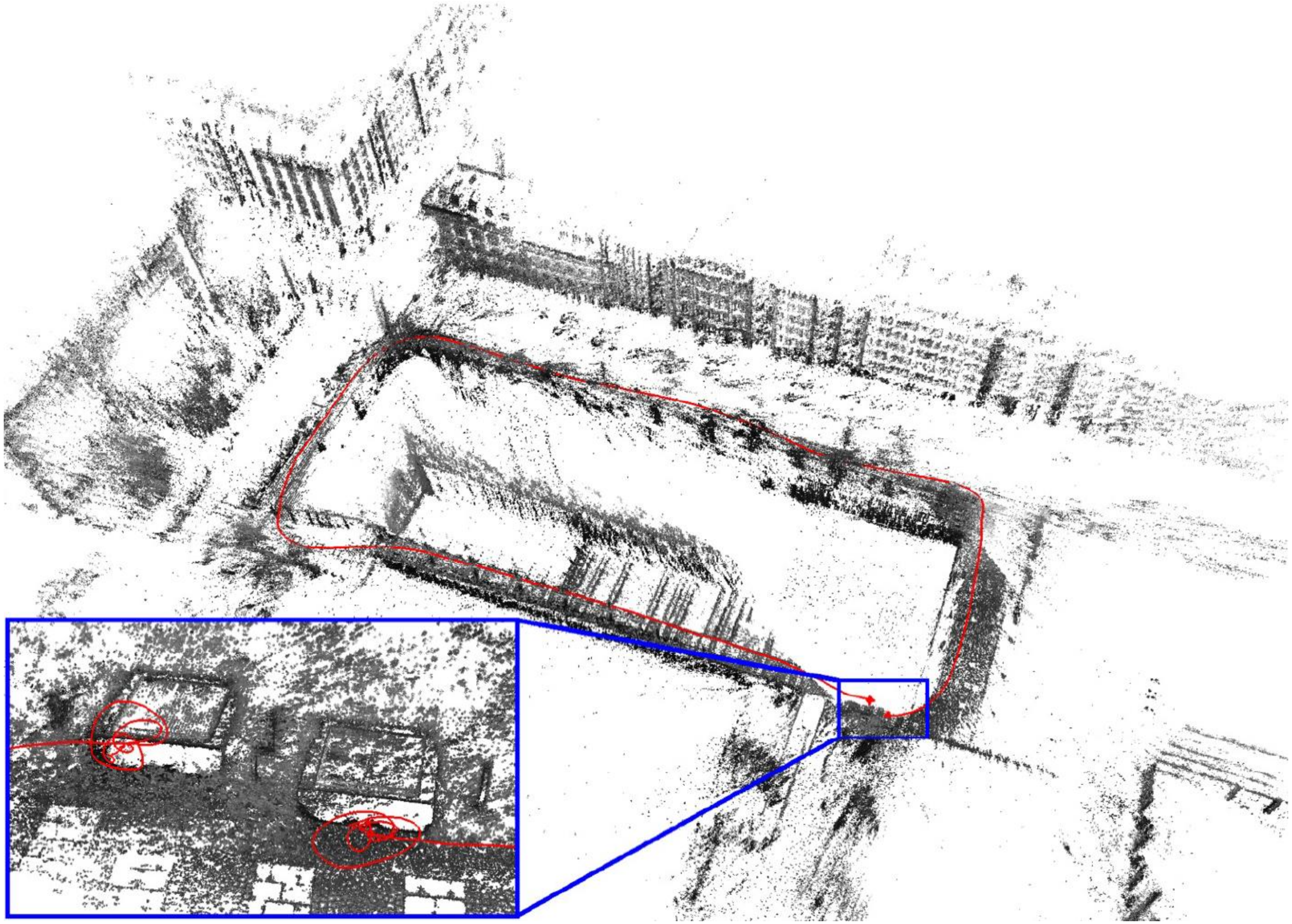
Felix Roessler

Technical University of Munich

04. December 2019



Direct Sparse Odometry



Direct Sparse Odometry

- Motivation
- Back-end: The Direct Sparse Model
- Front-end
- Results and Further Studies
- Conclusion

Direct Sparse Odometry

- **Motivation**
- Back-end: The Direct Sparse Model
- Front-end
- Results and Further Studies
- Conclusion

What is Direct Sparse Odometry?

Direct:

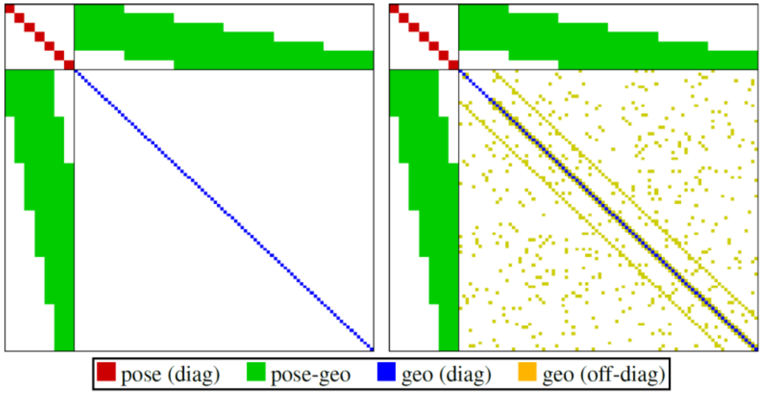
- Work directly with sensor input
- Optimize a photometric error

Sparse:

- Use and reconstruct only certain points
- No geometric prior

Odometry:

- Focus on accurate pose tracking
- Incremental estimation of the camera path
- Local optimization



Direct Sparse Odometry

- Motivation
- **Back-end: The Direct Sparse Model**
  - Main Characteristics
  - Error Function
- Front-end
- Results and Further Studies
- Conclusion

Back-end: Main Characteristics

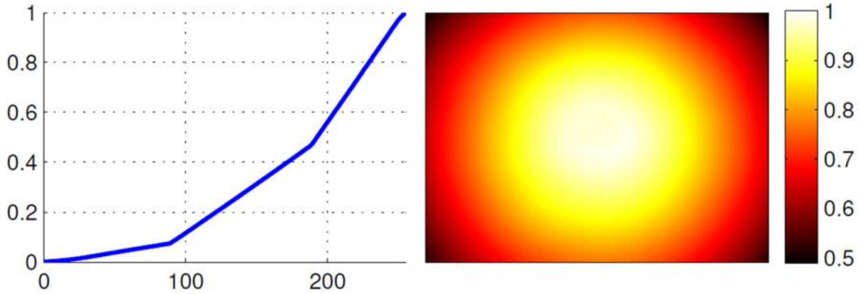
Continuous, local optimization over window of recent frames

Joint optimization for camera intrinsics, pose, and inverse depths

Photometric camera model:

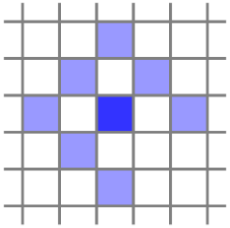
$$I_i(\boldsymbol{x}) = G\big(t_i V(\boldsymbol{x}) B_i(\boldsymbol{x})\big) \rightarrow I_i'(\boldsymbol{x}) \coloneqq t_i B_i(\boldsymbol{x}) = \frac{G^{-1}\big(I_i(\boldsymbol{x})\big)}{V(\boldsymbol{x})}$$

- $I_i(\boldsymbol{x})$: Observed pixel intensity
- $B_i(\boldsymbol{x})$: True Irradiance
- G: Response function (Gamma)
- V: Vignetting
- t_i: Exposure time



Inv. response function (G^{-1}), vignetting (V)

Back-end: Error Function



Weighted SSD error over small neighborhood of pixels

More information for each point due to contribution of multiple pixels

Error function:

$$E_{\boldsymbol{p}j} := \sum_{\boldsymbol{p}\in\mathcal{N}_{\boldsymbol{P}}} w_{\boldsymbol{P}} \left\| \left(I_j[\boldsymbol{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\boldsymbol{p}] - b_i\right) \right\|_{\gamma}$$

- $\mathcal{N}_{\boldsymbol{P}}$: Set of points contained in residual pattern
- $w_{\boldsymbol{P}}$: Gradient-dependent weight
- a_i, b_i: Affine brightness transformation parameters for unknown exposure times
- $\boldsymbol{p}'$: Projected point position: $\boldsymbol{p}' = \Pi_c(\boldsymbol{R}\Pi_c^{-1}(\boldsymbol{p}, d_{\boldsymbol{P}}) + \boldsymbol{t})$

Back-end: Error Function

$$E_{\boldsymbol{p}j} \;:=\; \sum_{\boldsymbol{p}\in\mathcal{N}_{\boldsymbol{P}}} w_{\boldsymbol{P}} \left\| \left(I_j[\boldsymbol{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\boldsymbol{p}] - b_i\right) \right\|_{\gamma}$$

Total photometric error:

$$E_{photo} \;:=\; \sum_{i\in\mathcal{F}} \sum_{\boldsymbol{p}\in\mathcal{P}_i} \sum_{j\in\mathrm{obs}(\boldsymbol{p})} E_{\boldsymbol{p}j}$$

Summation over:

- Points in residual pattern $\mathcal{N}_{\boldsymbol{P}}$
- Active frames $\mathcal{F}$
- Active points $\mathcal{P}_i$ in frame
- Frames $\mathrm{obs}(\boldsymbol{p})$ in which $\boldsymbol{p}$ is visible

Direct Sparse Odometry

- Motivation
- Back-end: The Direct Sparse Model
- **Front-end**
  - Tasks
  - Frame Management
  - Point Management
- Results and Further Studies
- Conclusion

Front-end: Tasks

Determine which points and frames to use for optimization

Initialize new parameters at high accuracy

Determine which points and frames to marginalize

Front-end: Frame Management

Tracking of new frames

- Align new frames with most recent keyframe
- Use image pyramid and constant motion model

Creation of new keyframes

- If a frame holds enough new information, create a new keyframe
- Check optical flow and exposure changes to measure information

Keyframe marginalization

- Drop frames with few points visible in most recent keyframe
- Keep keyframes well distributed in space

Front-end: Point Management

Select candidate points in new keyframe

- Well distributed
- Sufficiently high image gradient

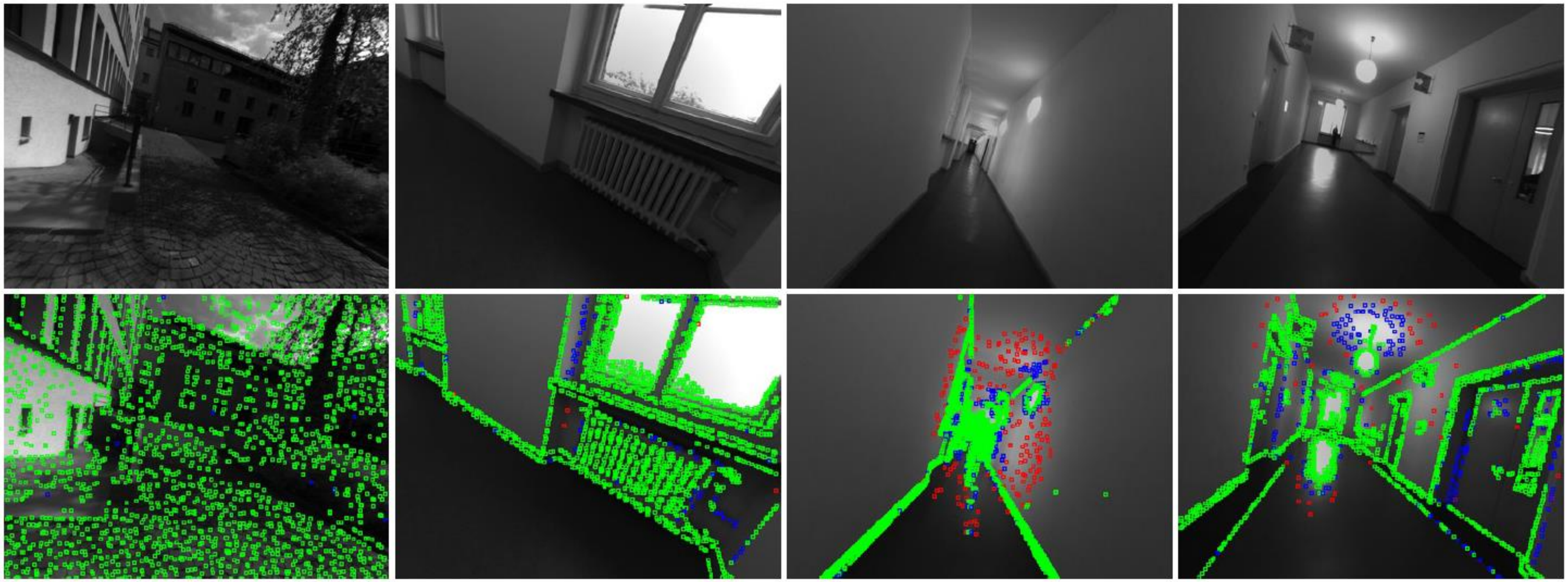
Initialize depths by tracking across subsequent frames

- Track points by search along epipolar line
- Discard points as outliers if there is no match

Add candidate points to optimization when needed

- Replace marginalized points with new ones to keep a fixed number of points
- Activate points to keep a good spatial distribution

Front-end: Point Management

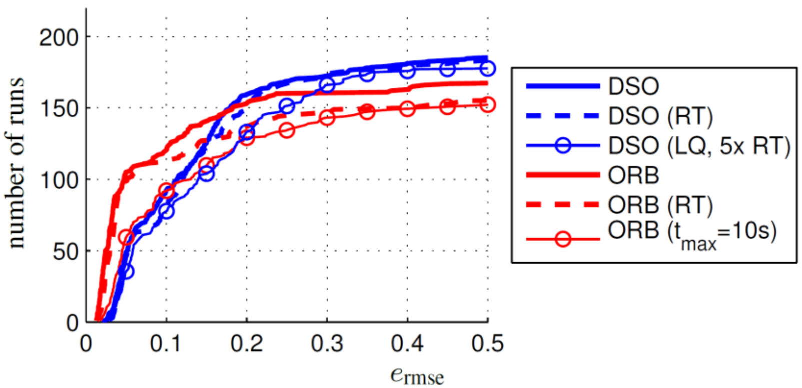


Candidate point selection: 2000 points in each keyframe

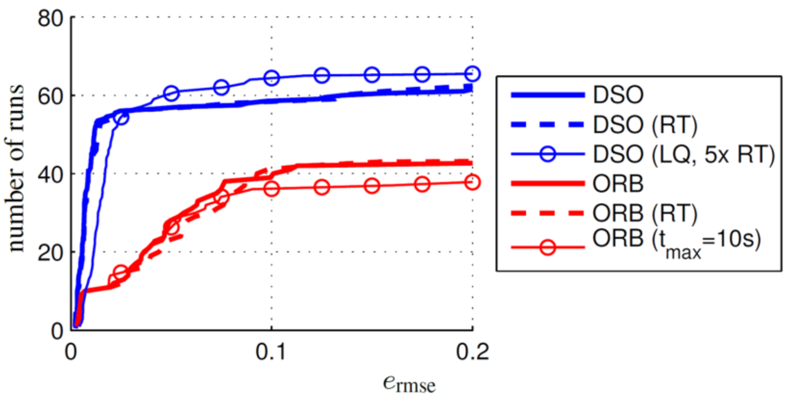
Direct Sparse Odometry

- Motivation
- Back-end: The Direct Sparse Model
- Front-end
- **Results and Further Studies**
  - Benchmark Performance
  - Parameter Studies
  - Noise Behavior
- Conclusion

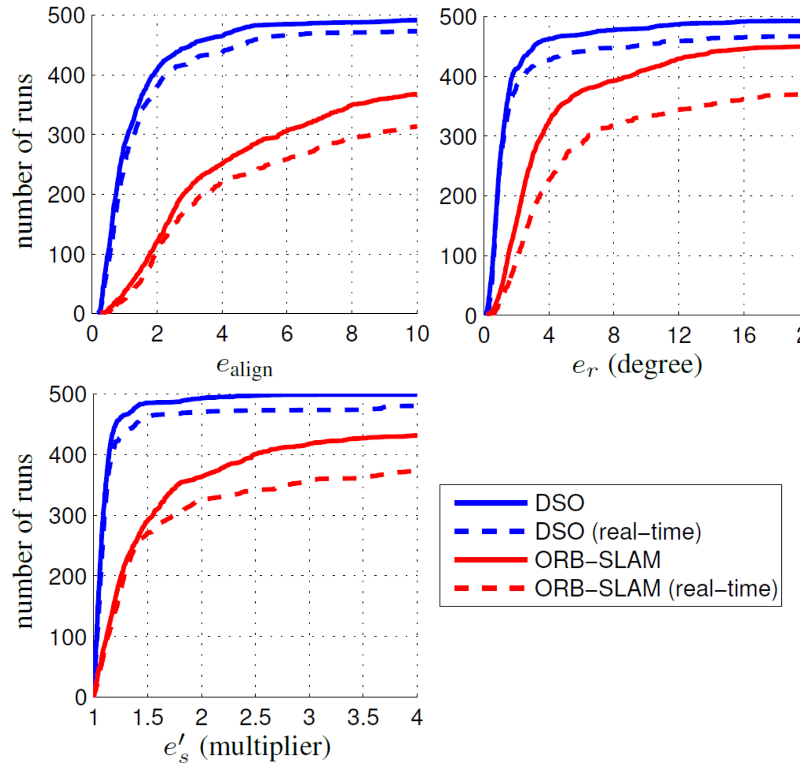
Results: Benchmark Performance



EuRoC MAV Dataset

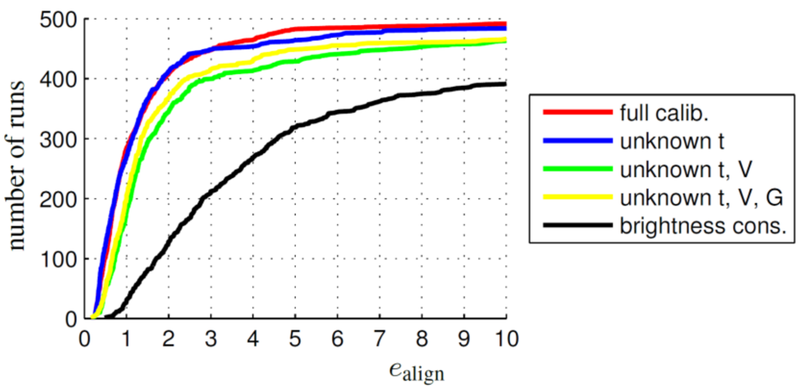


ICL_NUIM Dataset

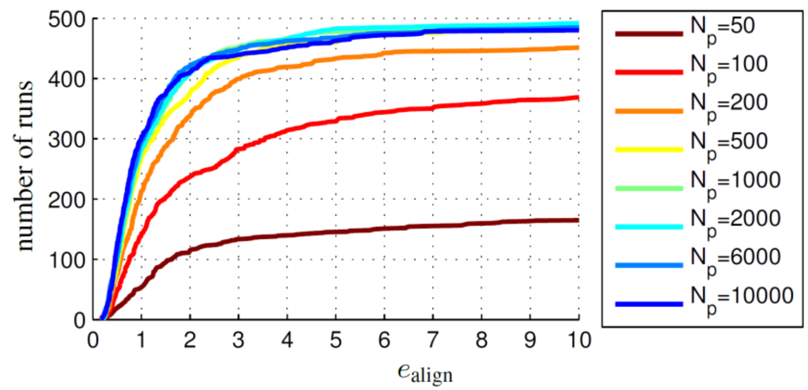


TUM-monoVO Dataset

Results: Parameter Studies

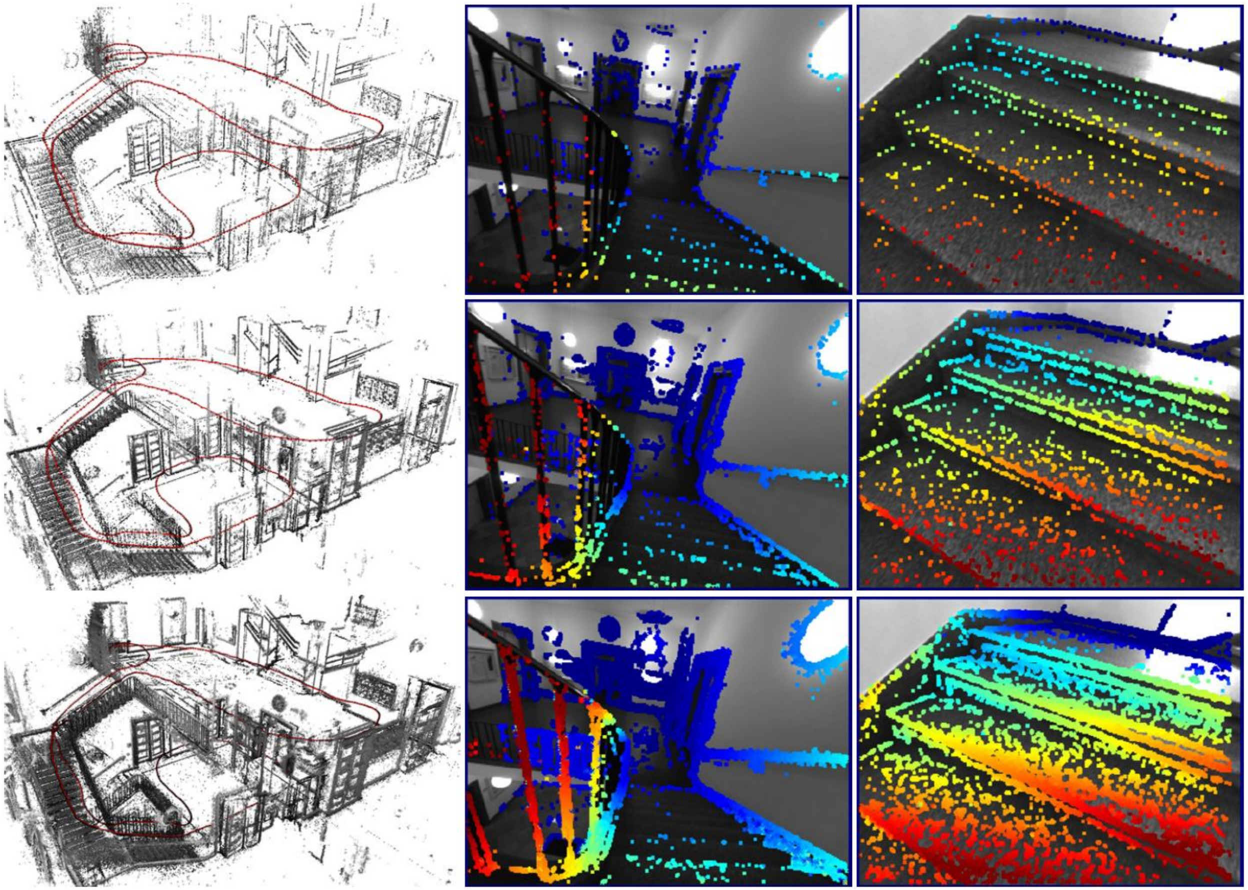


Photometric Calibration



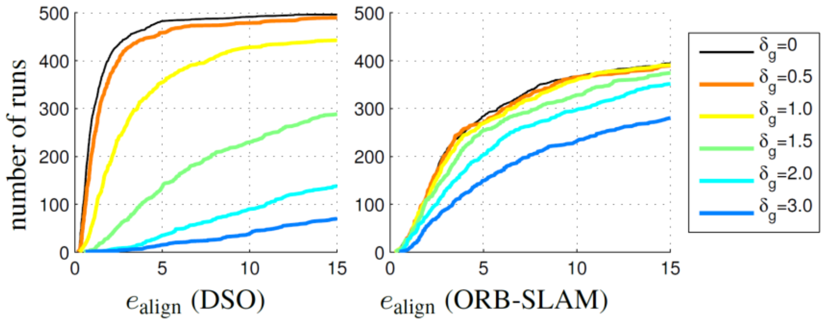
Number of Active Points

Results: Parameter Studies

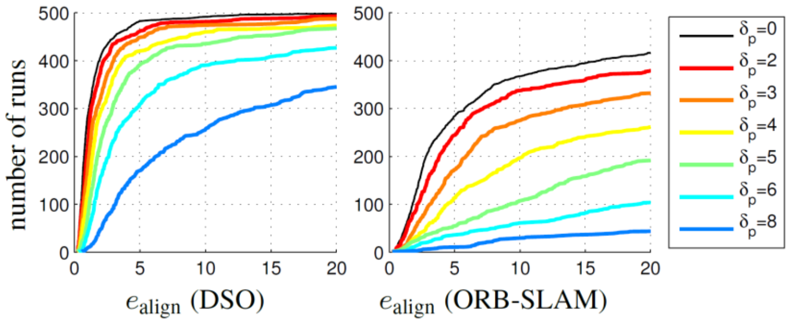


$\mathcal{N}_{\boldsymbol{P}} = 500$, 2000, and 10000 active points

Results: Noise Behavior



Geometric Noise



Photometric Noise

Direct Sparse Odometry

- Motivation
- Back-end: The Direct Sparse Model
- Front-end
- Results and Further Studies
- **Conclusion**

Conclusion

Advantages of DSO:

- Real-time capable on CPU
- Full photometric calibration included in optimization
- Sampling from all image regions, not only corners

Downsides of DSO:

- Sensitive to geometric noise (i.e. rolling shutter)
- Very accurate initialization of new frames required