Indirect Visual Odometry with Optical Flow

Kerem Yildirir

Poyraz Kivanc Karacam

Yigit Aras Tunali

Team 2

Visual-Inertial Mapping with Non-Linear Factor Recovery

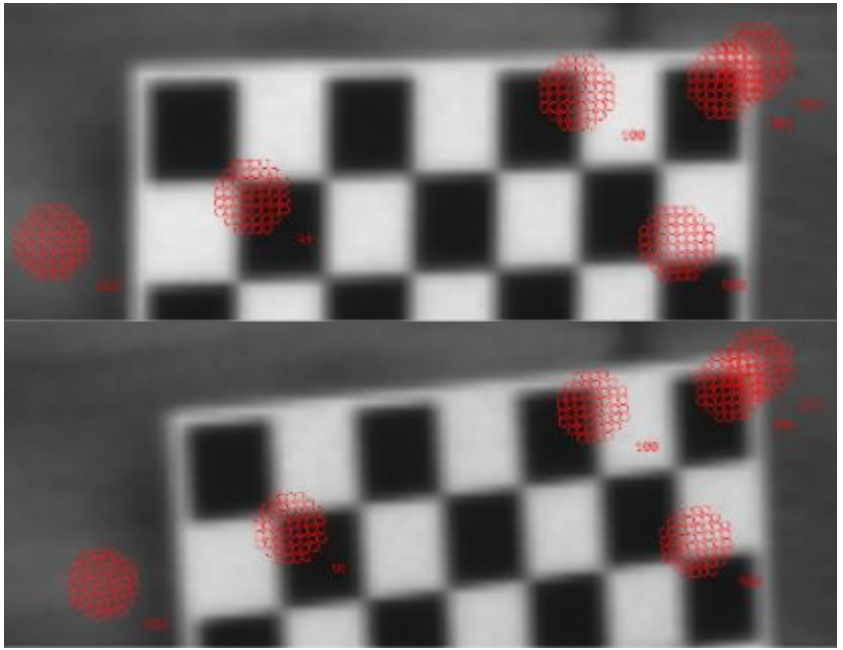
Vladyslav Usenko

Nikolaus Demmel

David Schubert

Jörg Stückler

Daniel Cremers

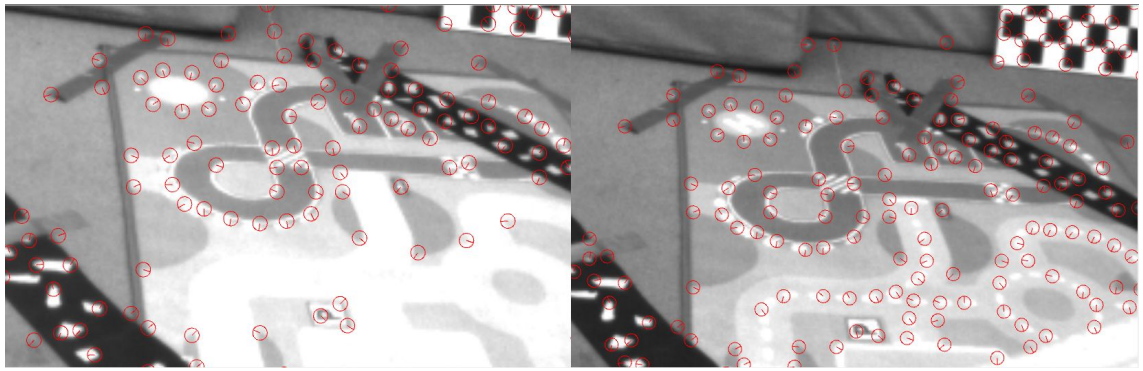


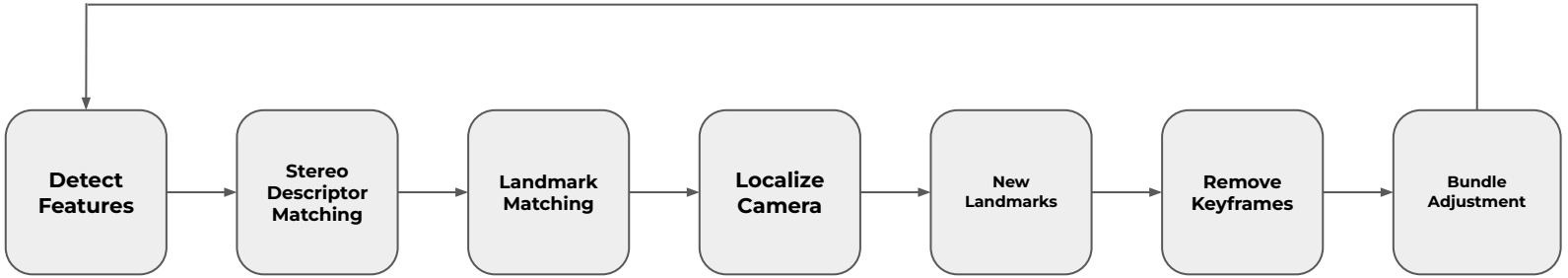
Optical flow feature tracking with BASALT

Outline

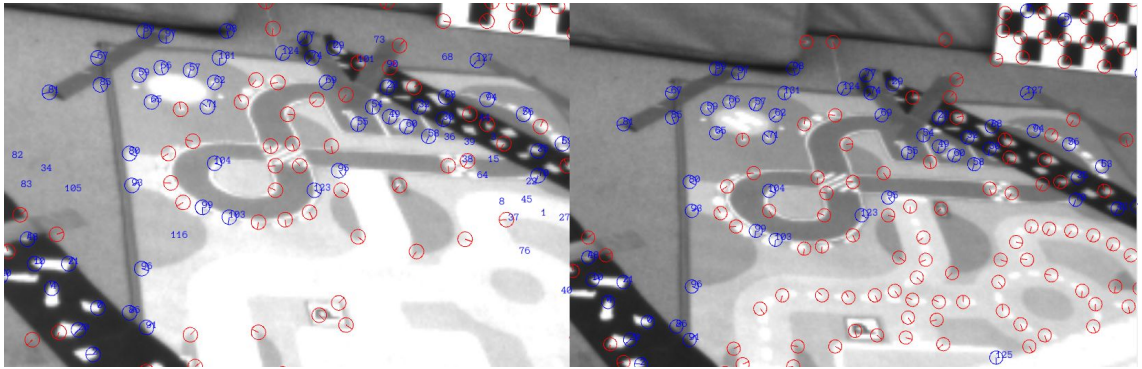
- Initial odometry recap and our extensions to the pipeline
- Optical flow and optimization explanation
- Analysis of results and conclusion

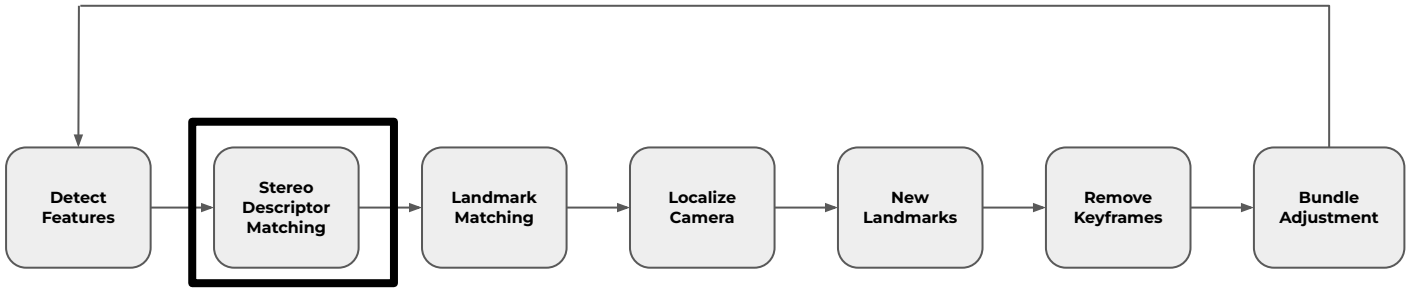
Odometry Pipeline



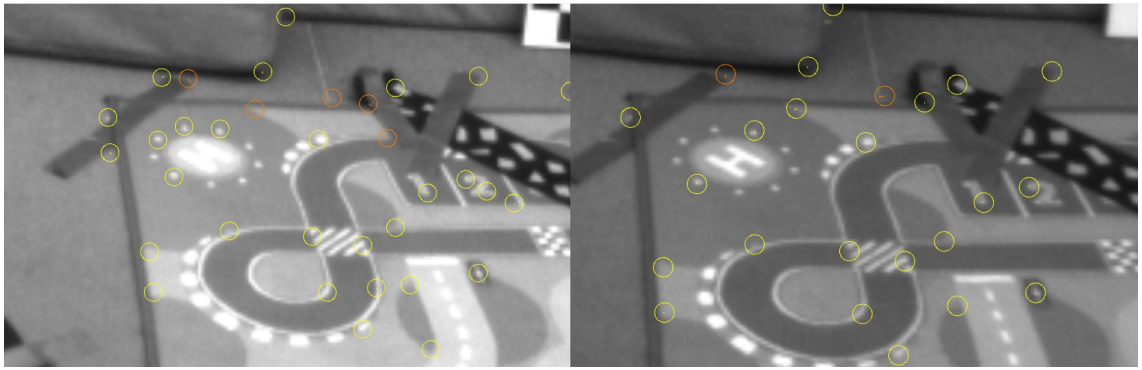


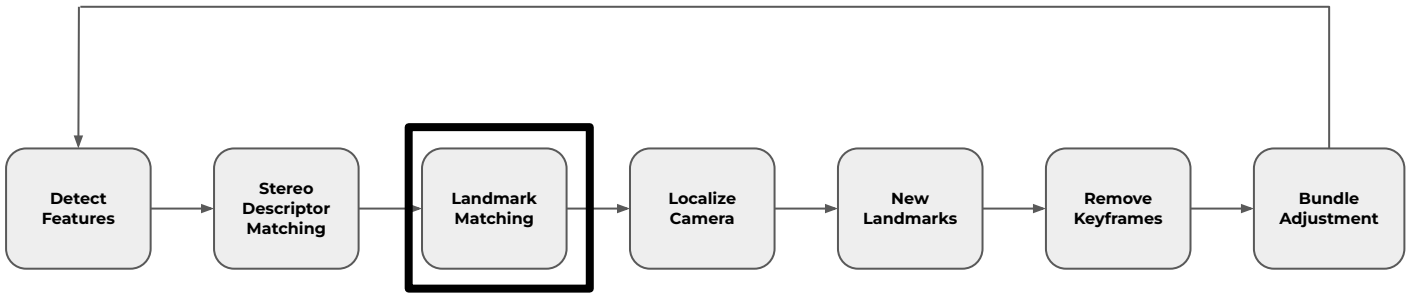
Odometry Pipeline



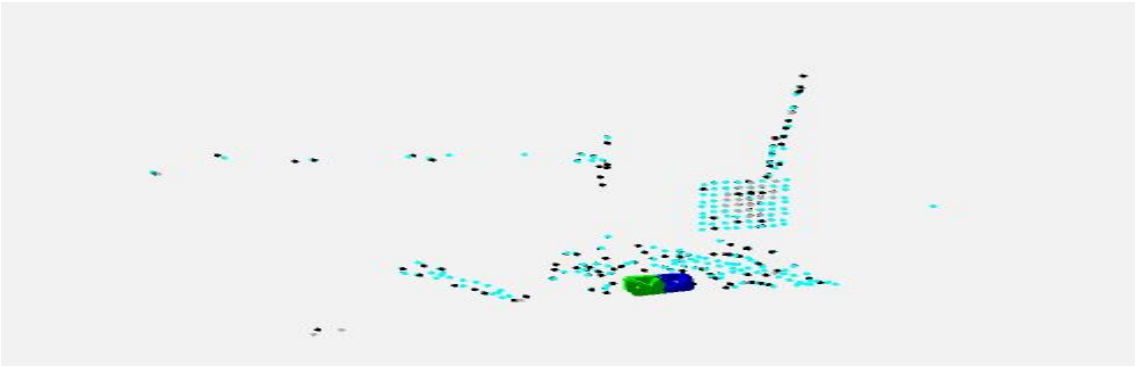


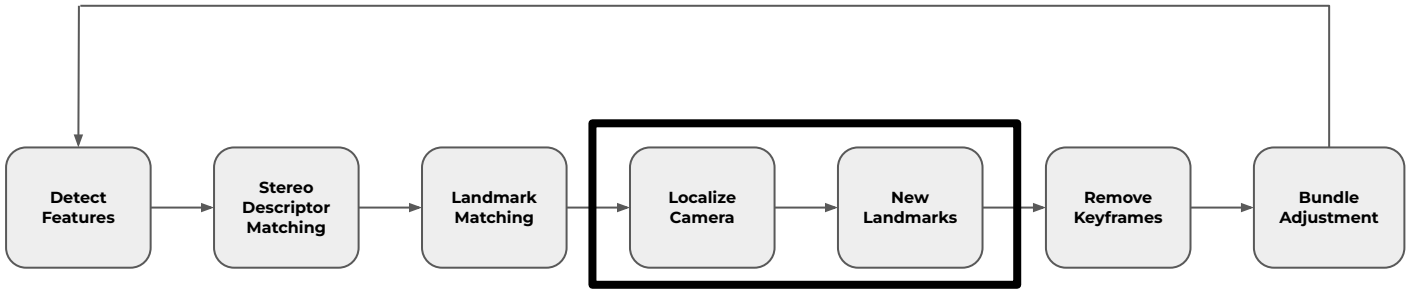
Odometry Pipeline



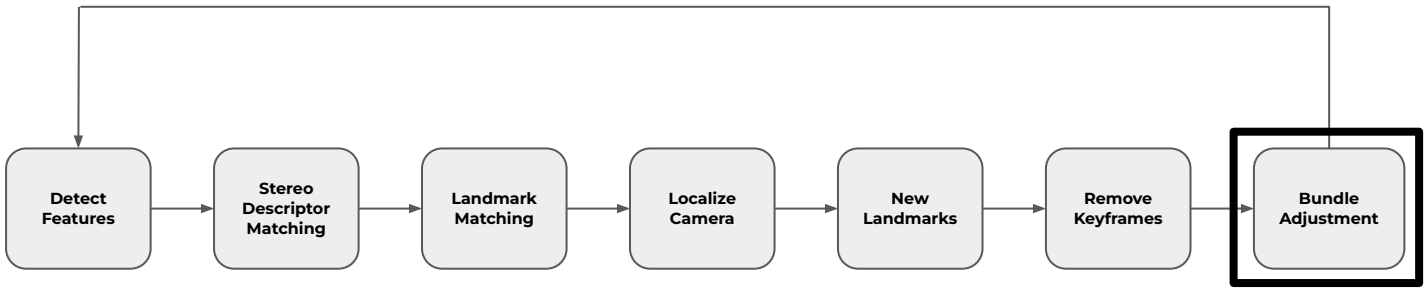


Odometry Pipeline

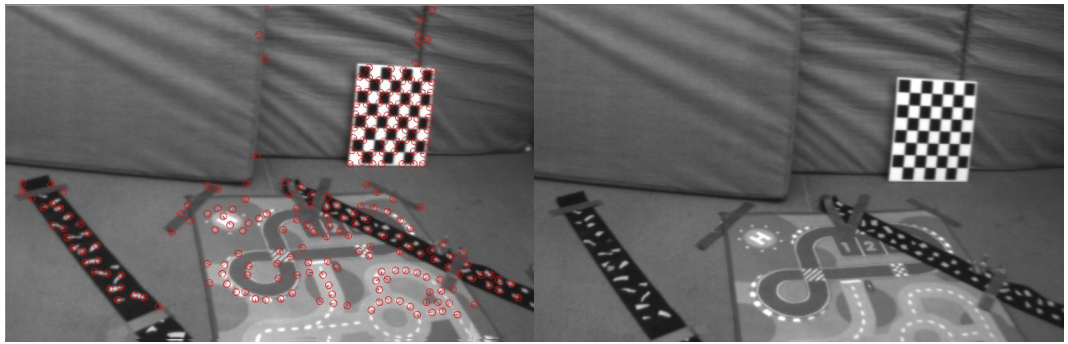


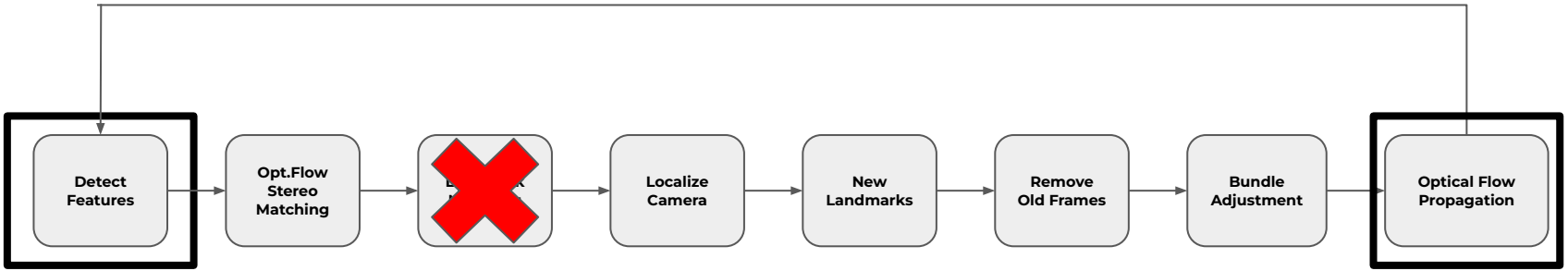


Odometry Pipeline



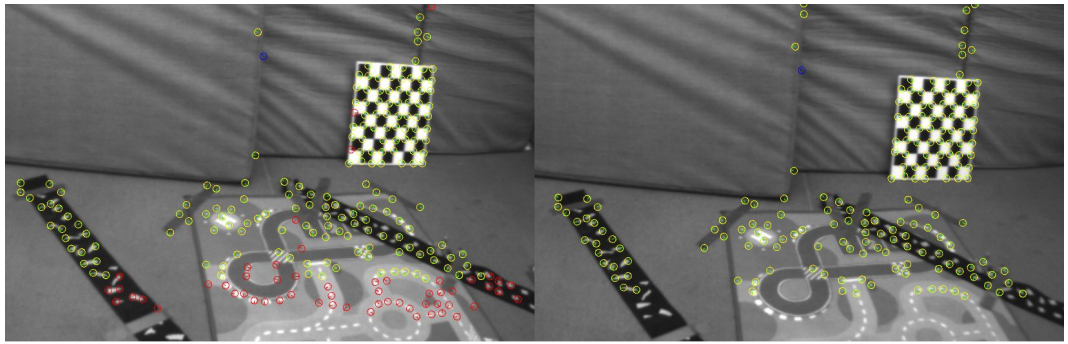
Our Method Summary

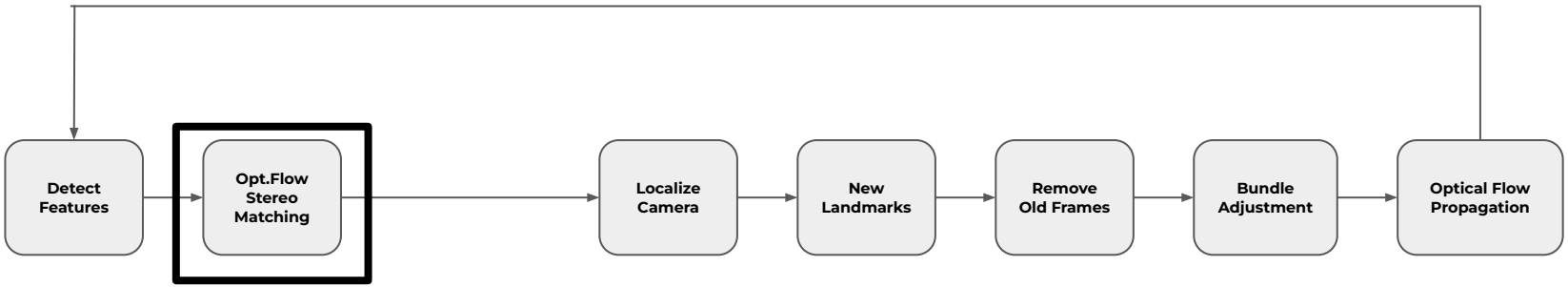




Only if not enough points!

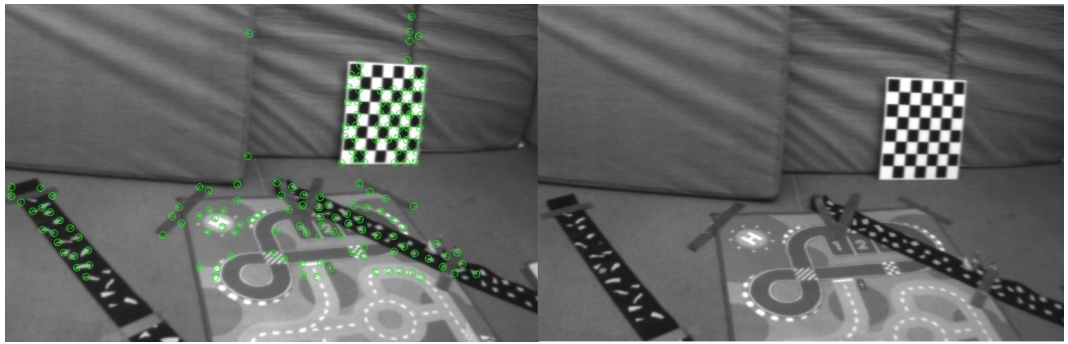
Our Method Summary

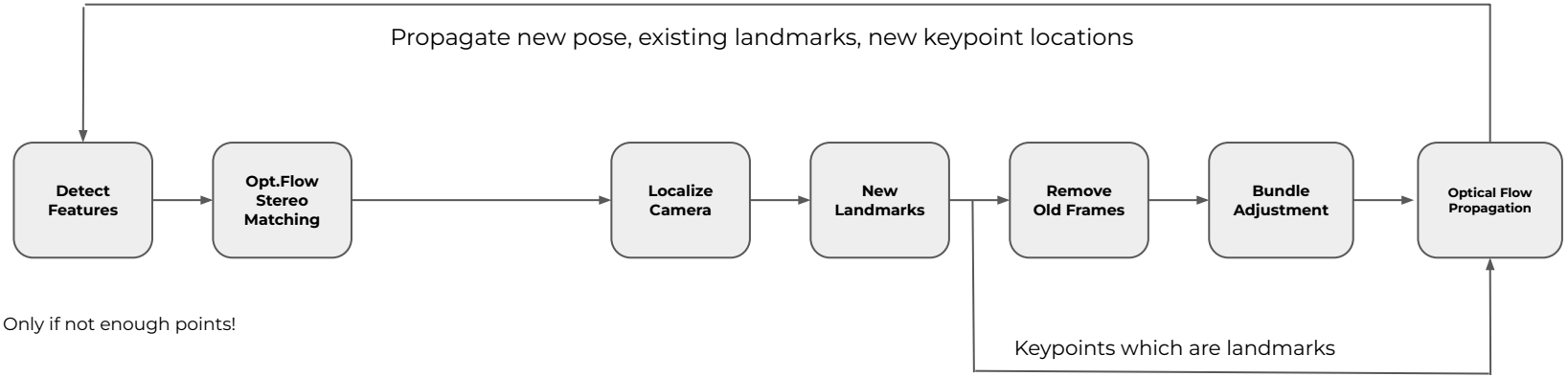




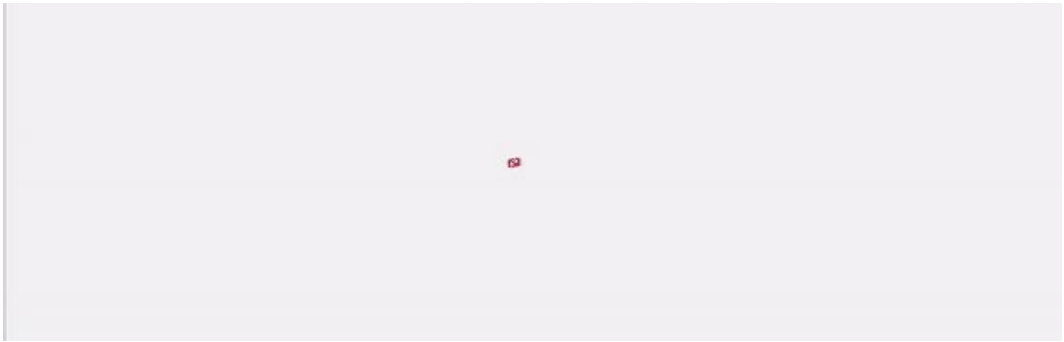
Only if not enough points!

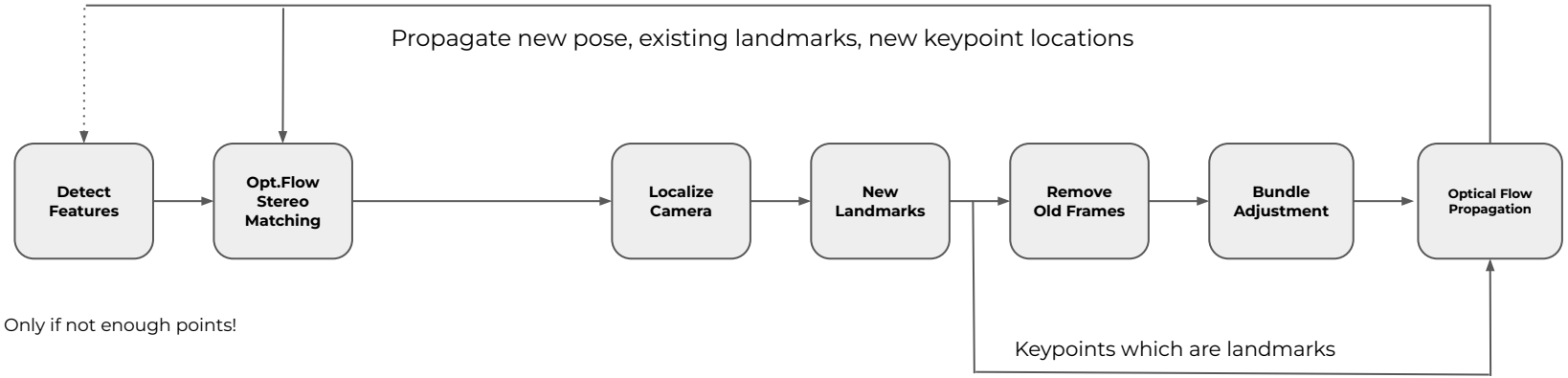
Our Method Summary



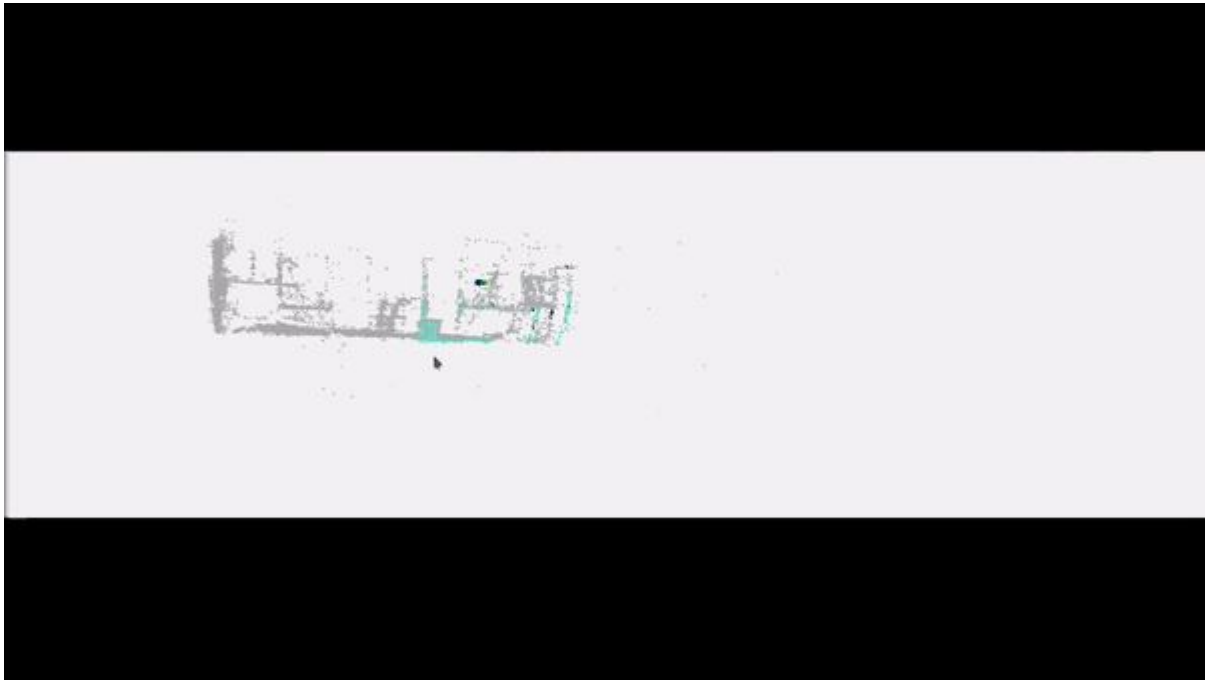


Our Method Summary





Generated 3D Map



Optical Flow

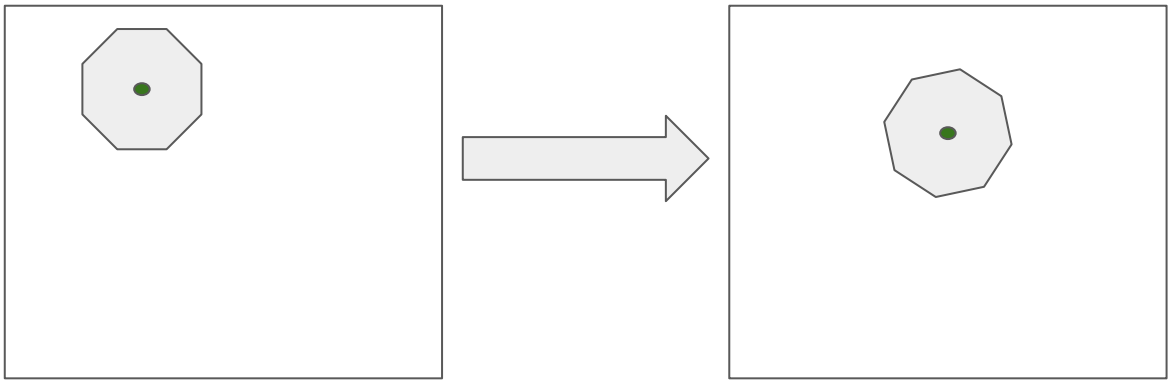
- Given two images:
  - Construct “Pyramids” from the images.
  - Track the points from image I to I’ on each level of the pyramids.
  - Track the points from I’ back to I.

- Patch Based Tracking.
- Find a warp that minimizes:

Tracking a point:

$$T \in SE(2)$$

$$r_i(\xi) = \frac{I_{t+1}(T\mathbf{x}_i)}{\overline{I_{t+1}}} - \frac{I_t(\mathbf{x}_i)}{\overline{I_t}} \; \forall \mathbf{x_i} \in \Omega$$

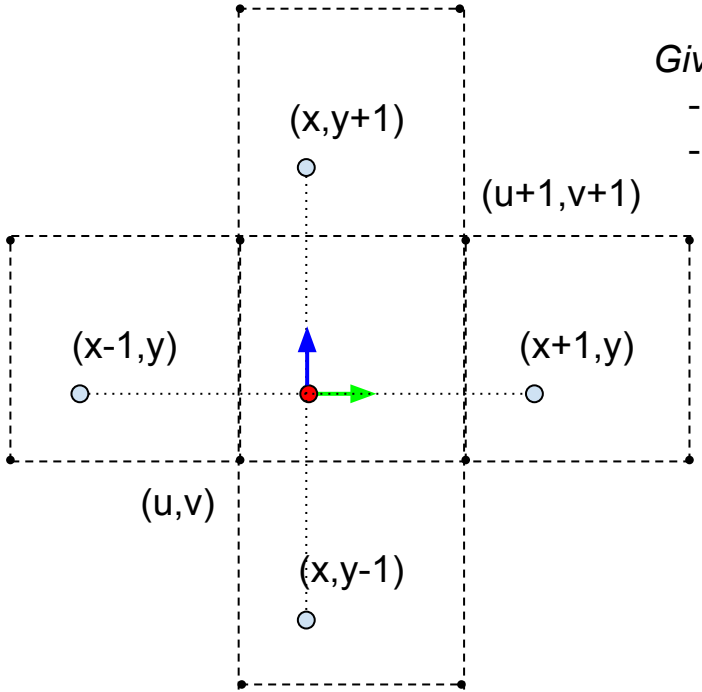


Optimizing the warp:

- Compute the gradients of each point in a patch.
- Incrementally update the warp until convergence.

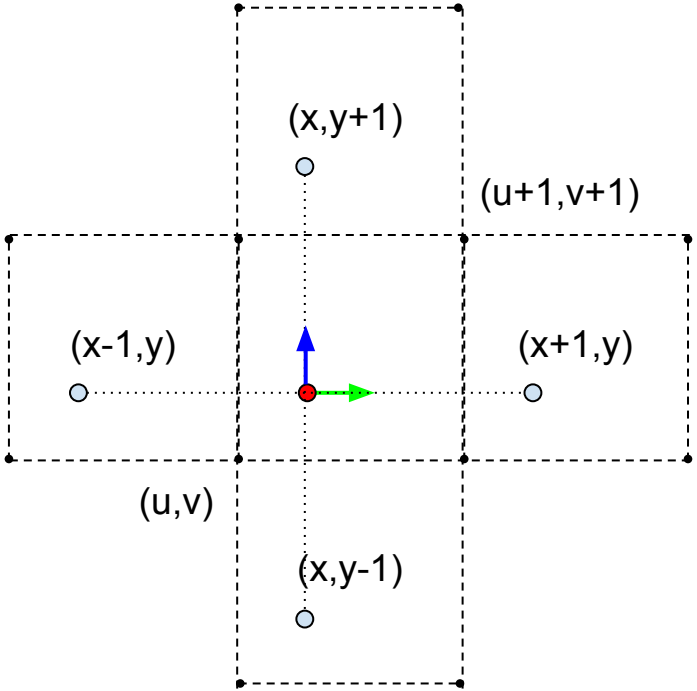
Point Gradients:

- Utilizing floating points.
- Bilinear Interpolation & Central Difference



Given point (x,y) *let*:

- u = ⌊x⌋
- v = ⌊y⌋

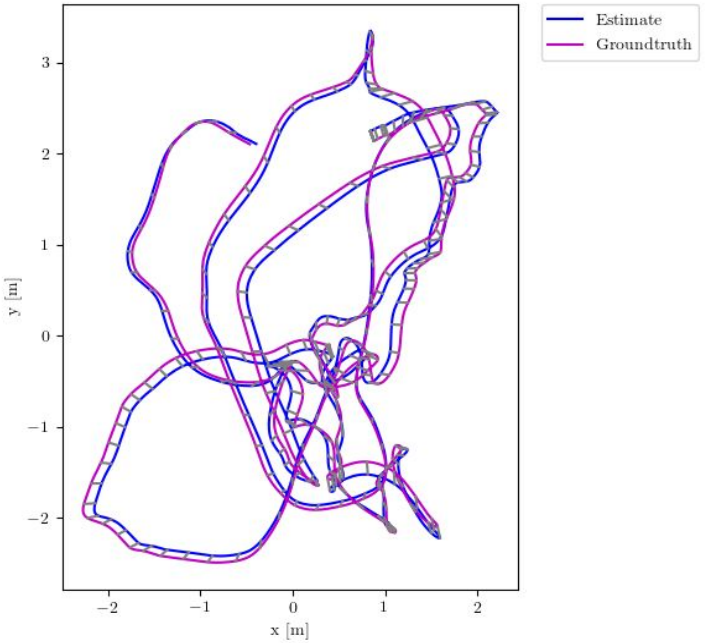

(x,y+1)
(u+1,v+1)
(x-1,y)
(x+1,y)
(u,v)
(x,y-1)

Computing the Increments:

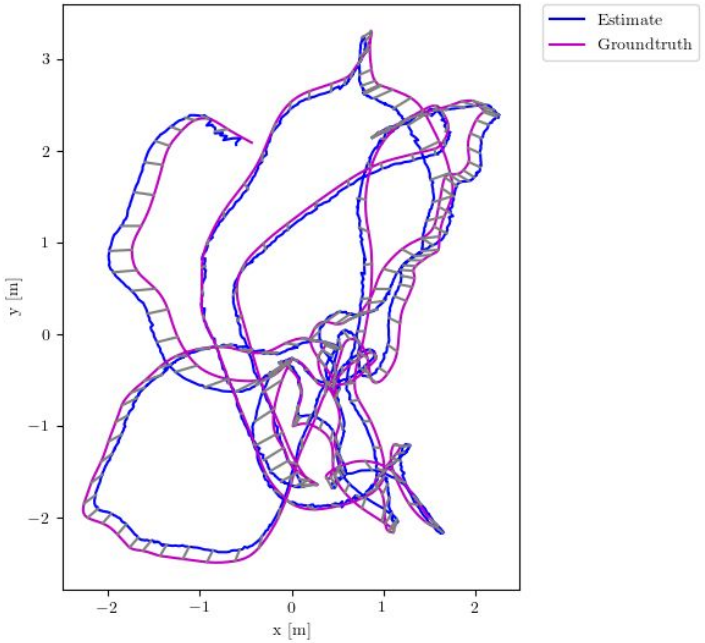
- Compute the Jacobian based on the gradients.
- Estimate the inverse Hessian using Cholesky Decomposition.
- Apply Gauss-Newton to the calculated residual $R^{t+1} = R^t exp(\xi)$
- The increment is applied to the rotation as:

Results and Experiments:

Results and Experiments: Trajectory Alignment

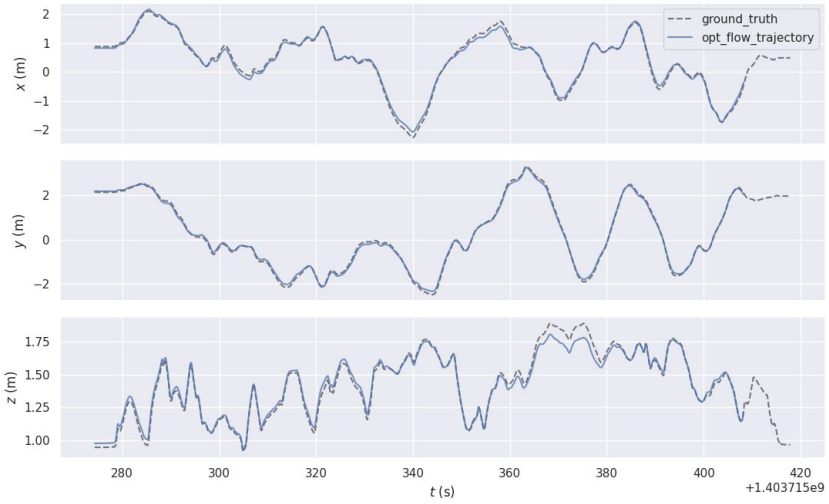


Ours

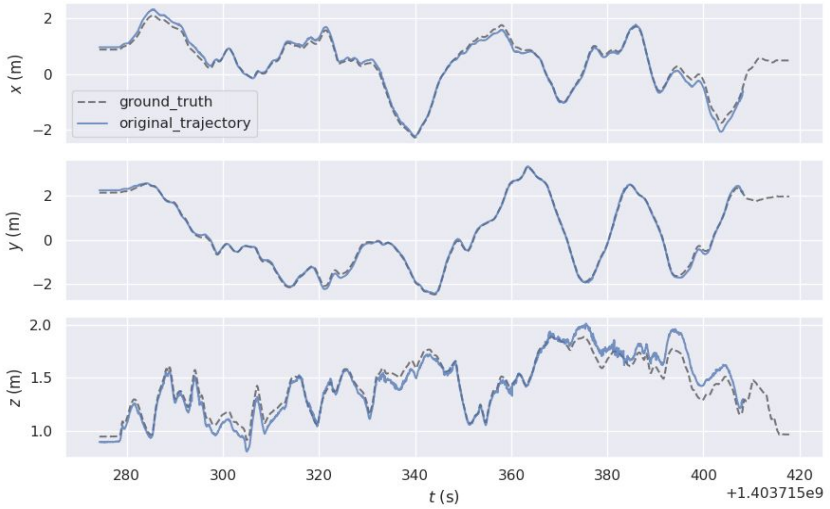


Original Odometry

Results and Experiments: XYZ difference in trajectory



Ours



Original Odometry

Relative Pose Error

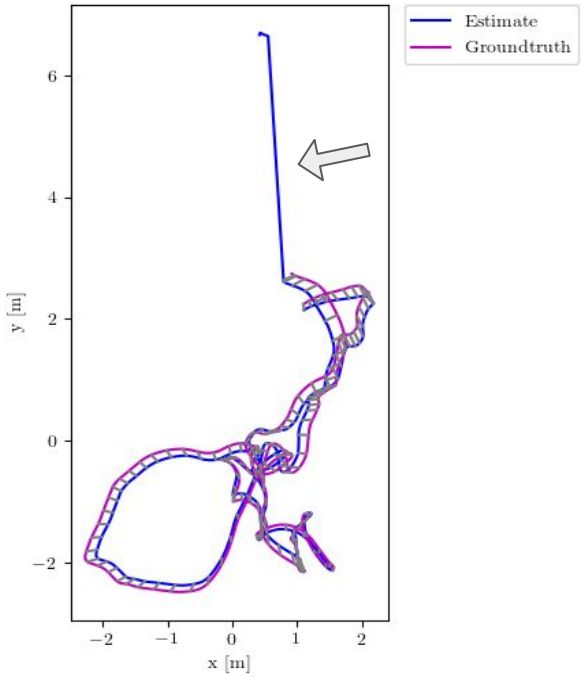
Method	rmse	mean	median	std	min	max
OpenCV	0.09	0.03	0.02	0.09	0.00	4.29
Ours	**0.03**	**0.03**	**0.02**	**0.02**	**0.00**	**0.07**
Original Odometry	0.04	0.03	0.03	0.02	0.00	0.17

Absolute Pose Error

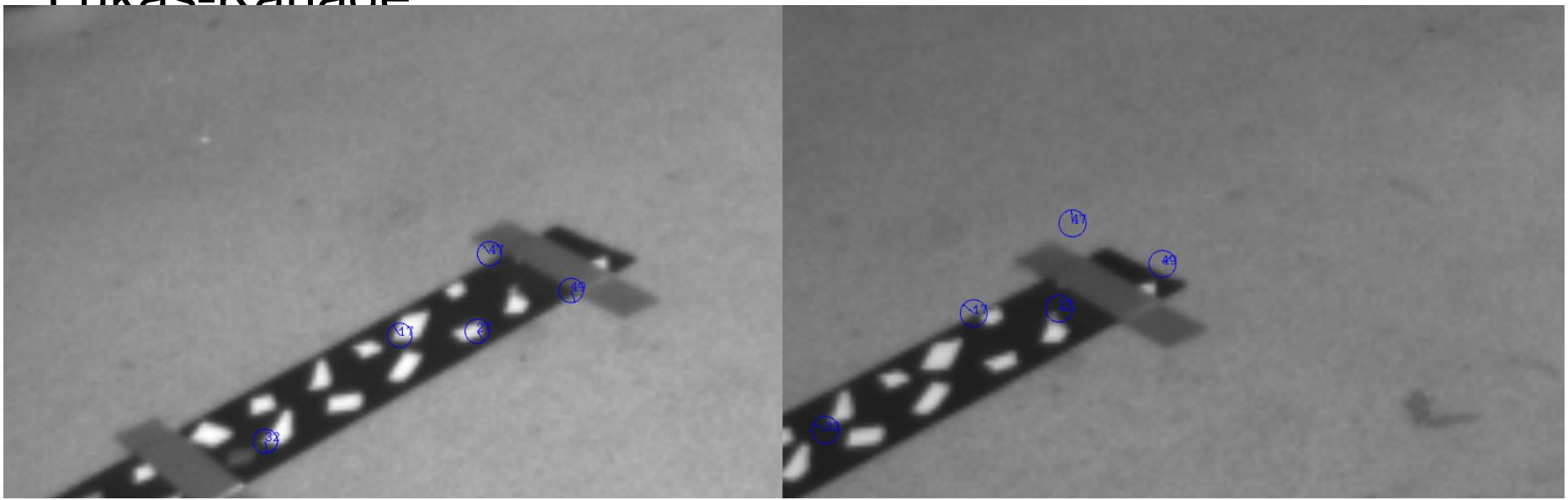
Method	rmse	mean	median	std	min	max
OpenCV	4.24	3.57	2.64	2.28	1.28	10.55
Ours	**0.12**	**0.11**	**0.11**	**0.04**	**0.01**	**0.21**
Original Odometry	0.16	0.14	0.14	0.07	0.01	0.37

Failure Case with OpenCV Lukas-Kanade

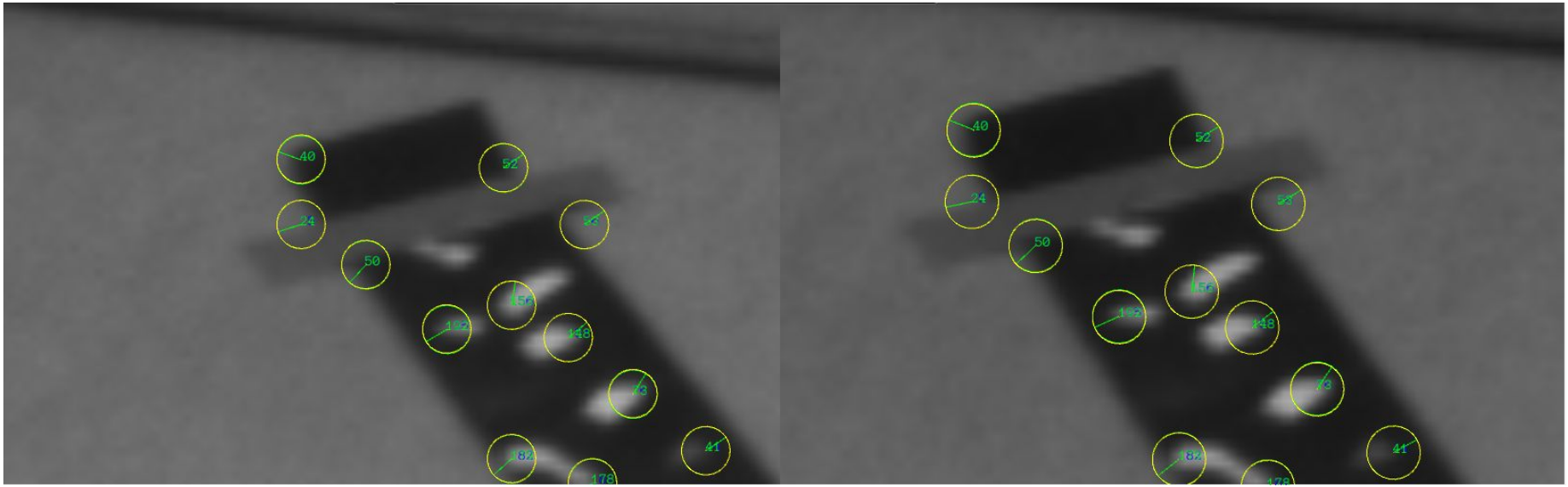
- When tracking fails, it is not able to recover
- For left to right optical flow, Lukas-Kanade didn’t work
- Hence we used Basalt approach for left to right



Stereo Matching with Optical Flow: OpenCV Lukas-Kanade



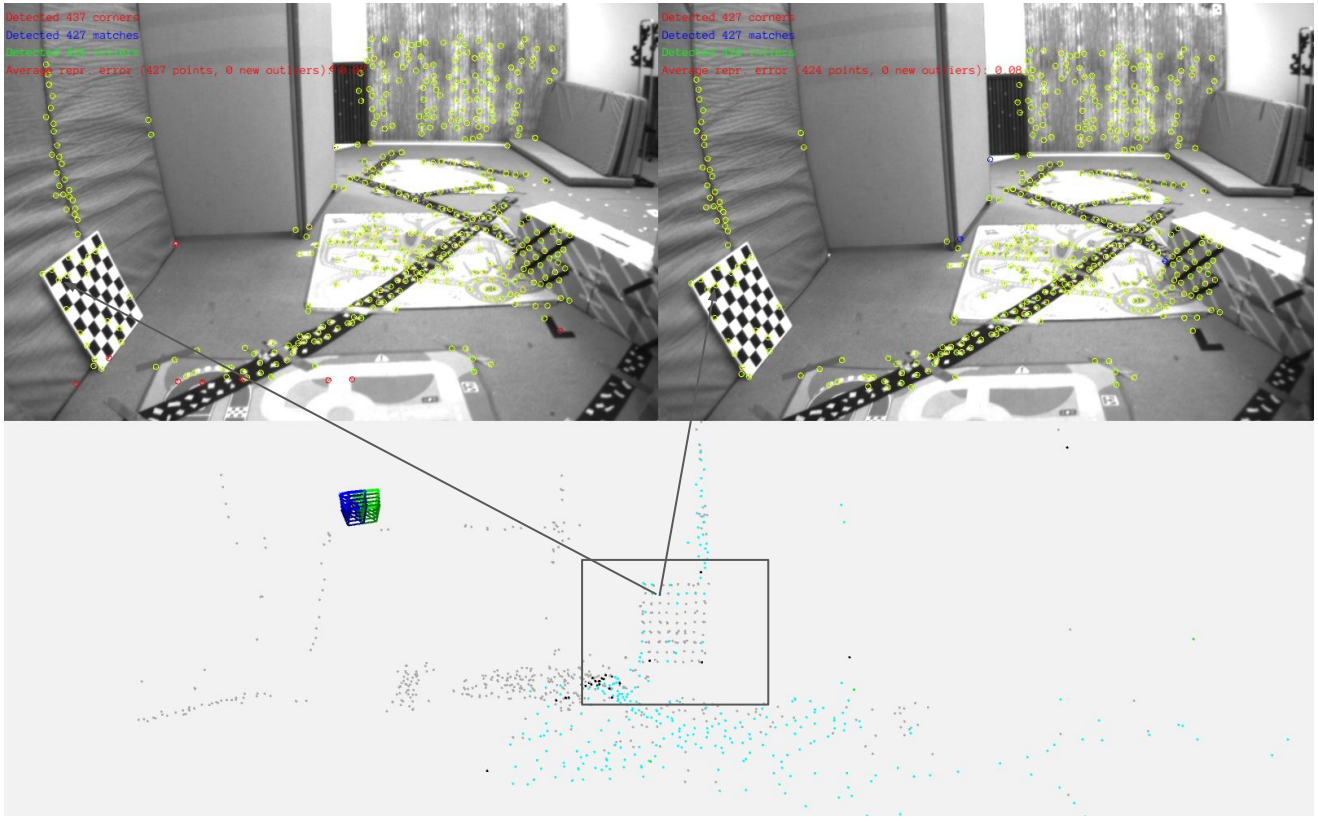
Stereo Matching with Optical Flow: Basalt



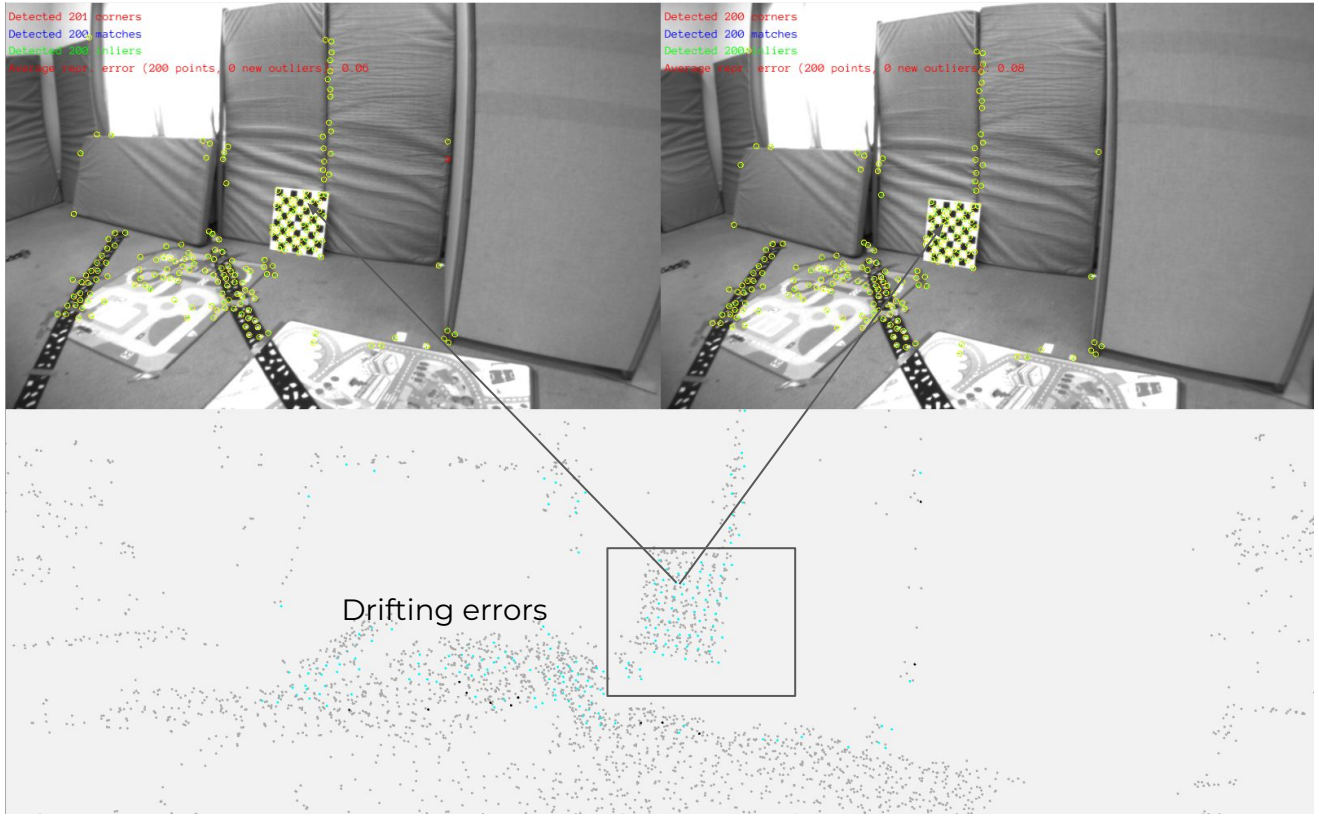
Shortcomings of the Method

- Frame to frame approach slower than Key Frame approach
- Detecting new keypoints slows down the system for a while
- Drift accumulation
- Optical flow fails in case of moves with big baseline (More robust with image pyramids)

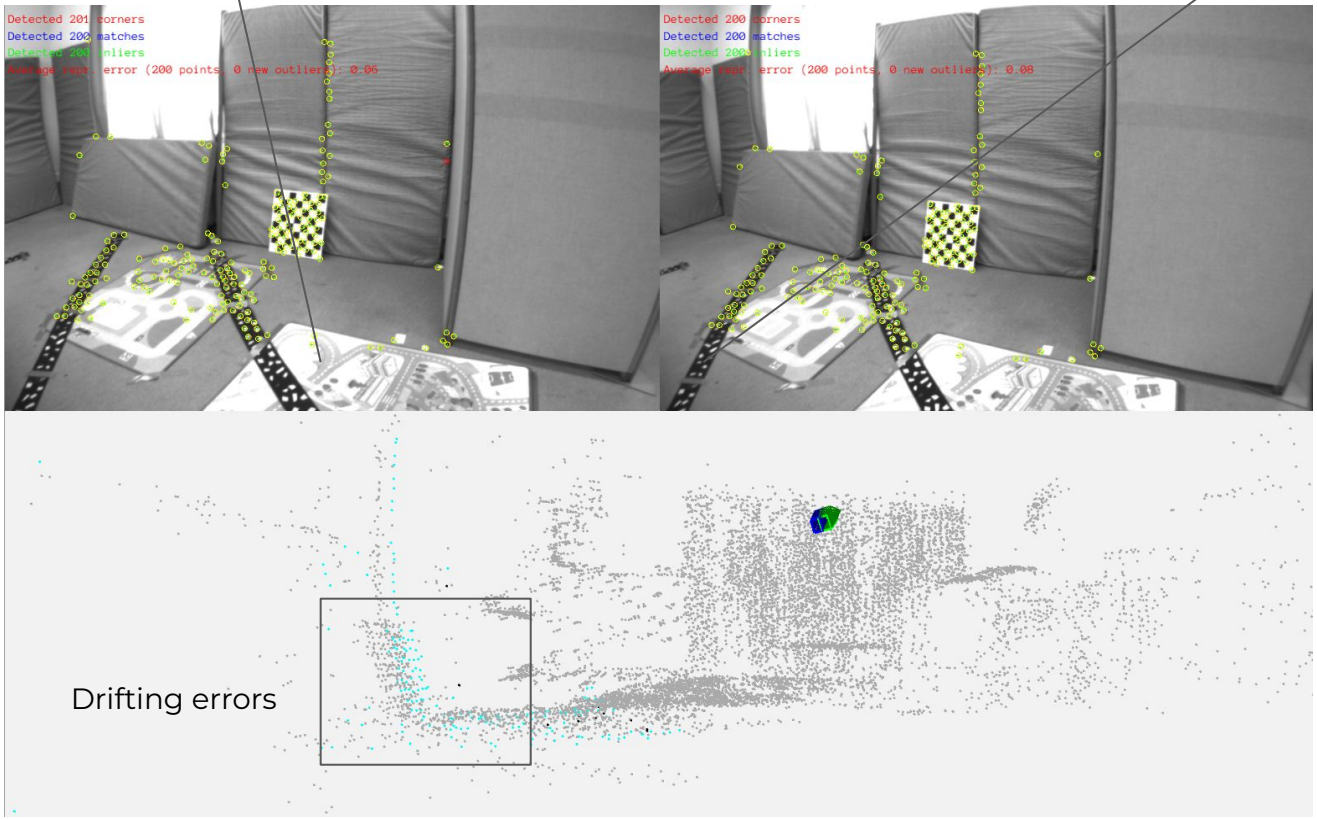
Effect of the Drift (Frame ~380)



Effect of the Drift (Frame ~ 1200)



Effect of the Drift (Frame ~2200)



Possible Extensions for Shortcomings

- Frame to frame will be slower inherently, possible parallelization of code
- Divide image in a grid and detect Keypoints separately for each grid
- For drift, loop closure is necessary

Conclusion

- Frame to frame drift is inevitable
- Good optical flow is more robust than descriptor matching
- Camera sequences with very big movements are not suitable

References

1. Visual-Inertial Mapping with Non-Linear Factor Recovery (V. Usenko, N. Demmel, D. Schubert, J. Stueckler and D. Cremers), In arXiv:1904.06504, 2019. https://arxiv.org/pdf/1904.06504
2. Equivalence and efficiency of image alignment algorithms (Baker, Simon, and Iain Matthews), In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 1. IEEE Computer Society; 1999, 2001. http://citeseerx.ist.psu.edu/ viewdoc/download?doi=10.1.1.70.20&rep=rep1&type=pdf
3. Basalt Codebase: https://gitlab.com/VladyslavUsenko/basalt
4. https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html
5. EVO https://github.com/MichaelGrupp/evo
6. RPG Toolbox https://github.com/uzh-rpg/rpg_trajectory_evaluation