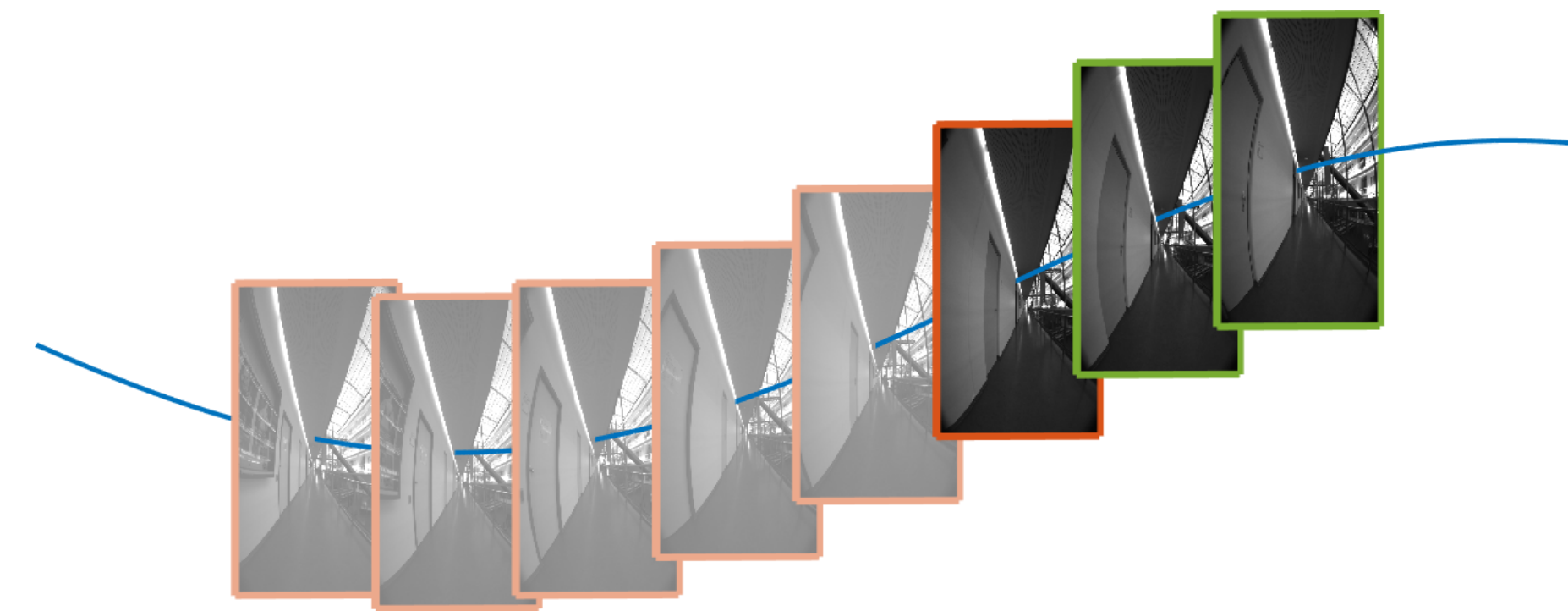


Idea

We propose a novel **square root formulation** for optimization-based **sliding-window estimators**. One of the components of this square root formulation is a **specialized QR decomposition** and we prove the **analytical equivalence** with marginalization based on **Schur complement**. Our QR marginalization naturally includes the case of **rank-deficient Jacobians**.

Sliding-window Bundle Adjustment



For optimization-based visual odometry, it is common to optimize a window of recent frames (green). Through marginalization, information from older frames (light orange) is retained. We show how to marginalize a frame (orange) when the energy is stored in square root form, which proves to be numerically advantageous.

Optimized energy with marginalization prior

Sliding-window energy, consisting of active residuals r_a and marginalization term E_m :

$$E_{sw}(x) = \frac{1}{2} \|r_a(x)\|^2 + E_m(x)$$

The marginalization cost E_m is a quadratic function in Δx , a perturbation from the current state x . While it is conventionally stored in Hessian form, we propose to equivalently use the Jacobian form, where J_m is a square root of H_m , i.e., $H_m = J_m^T J_m$:

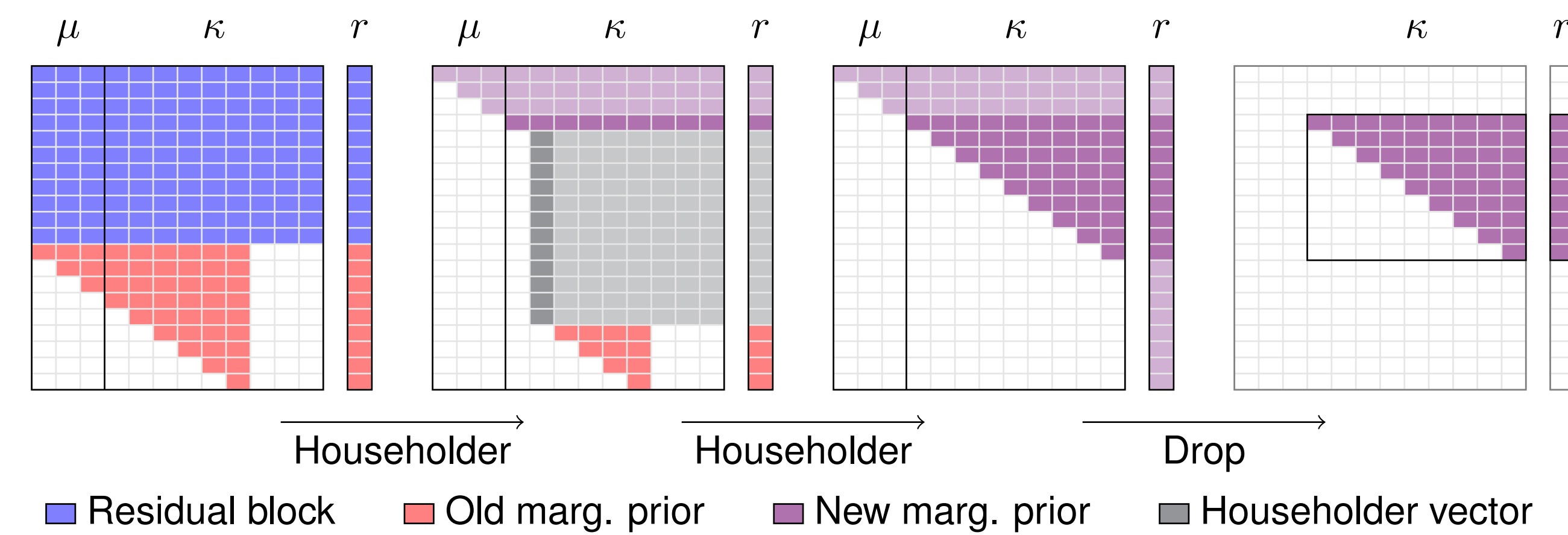
$$E_m(x) = \frac{1}{2} \Delta x^T H_m \Delta x + b_m^T \Delta x + \text{const} = \frac{1}{2} \|r'_m + J_m \Delta x\|^2$$

For optimization and marginalization, the non-linear least squares sliding-window energy $E_{sw}(x)$ is linearized:

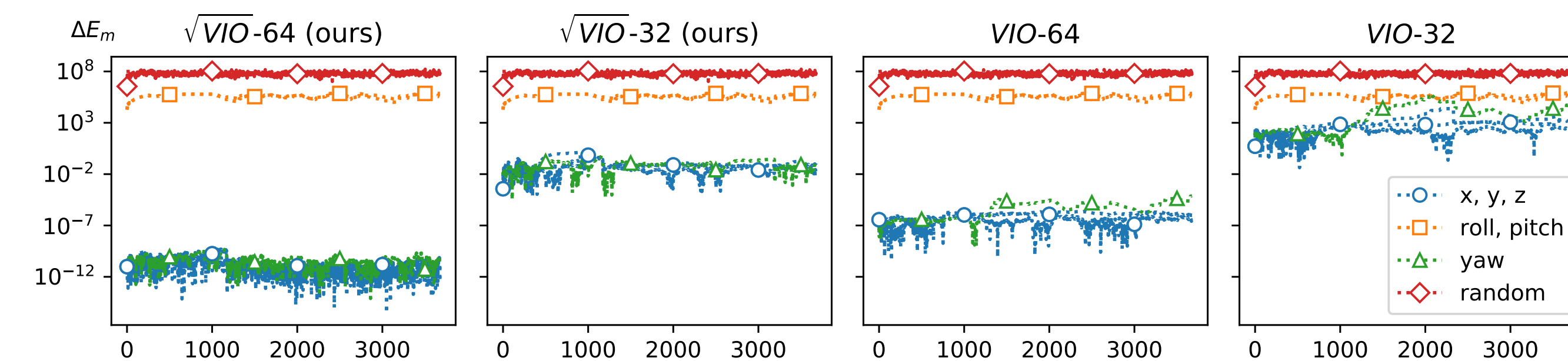
$$E_{sw,lin}(\Delta x) = \frac{1}{2} \|r + J \Delta x\|^2$$

Frame marginalization in square root form

Sorting marginalized states to the left and performing in-place QR decomposition with Householder reflections allows to easily extract the compact new marginalization prior.



Nullspace and eigenvalue analysis highlights numerical stability of square root form

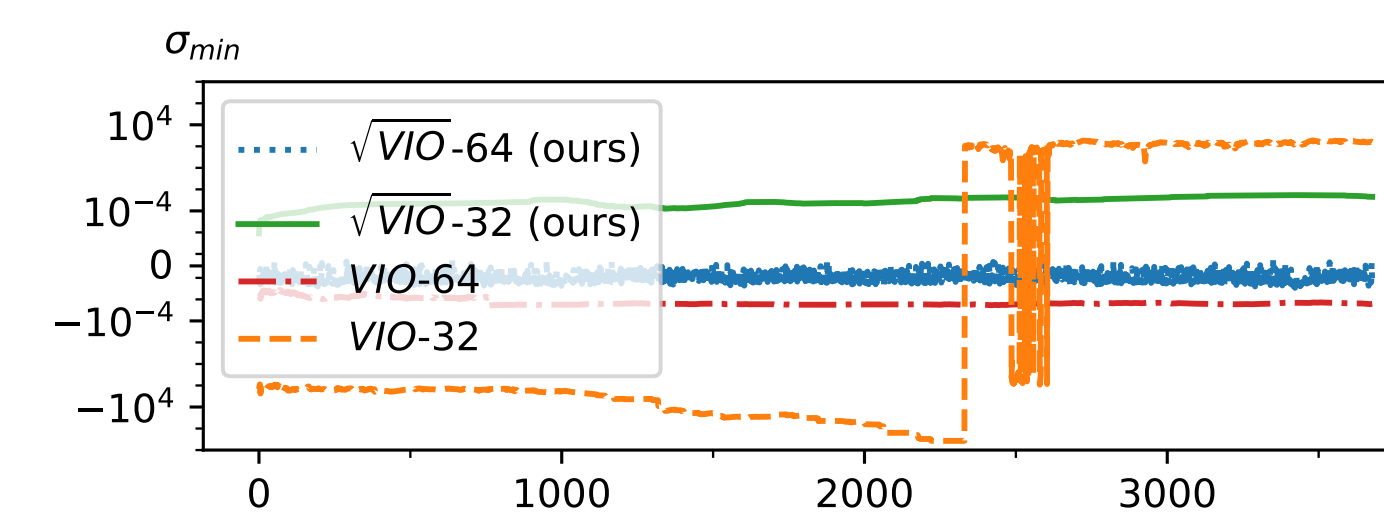
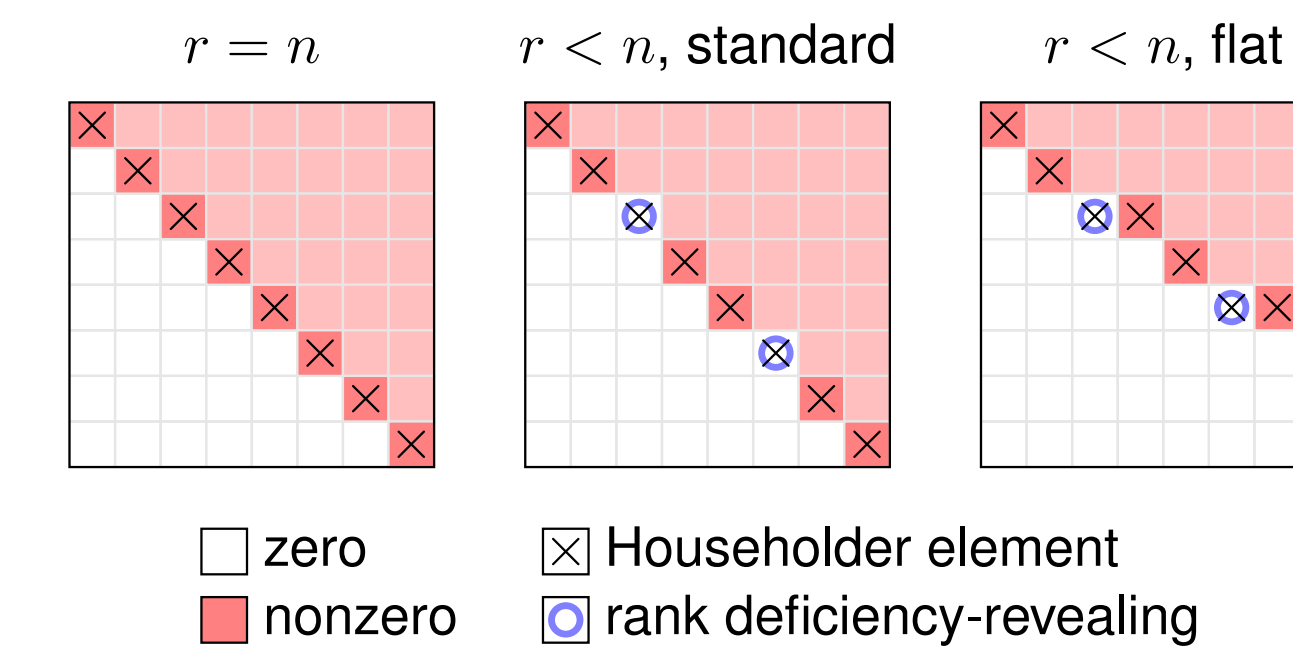


Absolute trajectory error for VIO

	$\sqrt{VIO-64}$	$\sqrt{VIO-32}$	VIO-64	VIO-32
eurocMH01	0.093	0.093	0.093	0.991
eurocMH02	0.048	0.048	0.048	0.048
eurocMH03	0.051	0.051	0.051	x
eurocMH04	0.109	0.109	0.109	x
eurocMH05	0.137	0.137	0.137	x
eurocV101	0.043	0.043	0.043	0.043
eurocV102	0.048	0.048	0.048	0.048
eurocV103	0.058	0.058	0.058	x
eurocV201	0.037	0.037	0.037	0.037
eurocV202	0.053	0.053	0.053	x
tumvi-corr1	0.300	0.300	0.300	x
tumvi-corr2	0.426	0.426	0.426	x
tumvi-mag1	1.456	1.457	1.456	x
tumvi-mag2	0.908	0.920	0.908	x
tumvi-room1	0.102	0.102	0.102	0.104
tumvi-room2	0.071	0.071	0.071	x
tumvi-slides1	0.310	0.310	0.310	x
tumvi-slides2	0.759	0.759	0.759	x

Flat QR for rank-deficient Jacobians

Our specialized, pivoting-free QR decomposition naturally includes the case of rank-deficient Jacobians. When encountering zeros on the diagonal we continue with the Householder element on the same row for the next column, resulting in a flat decomposition.



Ablation study

Different combinations of optimization and marginalization techniques, and floating-point precision for \sqrt{VIO} on EuRoC. All variants store the marginalization prior in square root form. The proposed square root marginalization *NS+QR* is deciding for good accuracy in single precision, while the square root optimization *NS+LDLT* leads to best runtime.

opt.	proposed		ablation study				
	NS+LDLT	NS+QR	SC+LDLT	NS+LDLT	SC+LDLT	SC+LDLT	
marg.	NS+QR	NS+QR	NS+QR	SC+SC	SC+SC	SC+SC	
precision	64	32	64	32	64	32	
ATE [m]	0.068	0.068	0.068	0.068	0.232	0.068	0.211
real-time	6.9x	8.2x	5.0x	5.6x	7.1x	7.9x	5.2x
t total [s]	17.9	14.9	24.4	21.8	17.4	15.5	23.7
t opt [s]	14.4	11.4	22.2	20.3	14.4	11.5	22.1
t marg [s]	1.6	1.5	1.6	1.3	1.4	1.4	1.3

Conclusion

The proposed square root formulation enables optimization-based VO and VIO with **single-precision** floating point computations **without loss of accuracy**. On several real-world datasets, we systematically analyze the effect of using nullspace projection combined with our specialized QR decomposition on runtime and accuracy. The proposed square root estimator is 23% faster than the baseline and **36% faster** when additionally switching to single precision.

Trajectory comparison



The conventional baseline *VIO-64* works well with double-precision floats, but fails in single precision (*VIO-32*). In contrast, the proposed square root estimator $\sqrt{VIO-32}$ even in single precision retains the same accuracy.

Acknowledgements

This work was supported by the ERC Advanced Grant *SIMULACRON* and by the German Science Foundation Grant CR 250/20-1 *Splitting Methods for 3D Reconstruction and SLAM*.

Code & Contact

Code is available open-source:
<https://go.vision.in.tum.de/rootvo>

Contact: Nikolaus Demmel
 nikolaus.demmel@tum.de

