

Square Root Marginalization for Sliding-Window Bundle Adjustment

Supplementary Material

Nikolaus Demmel David Schubert Christiane Sommer Daniel Cremers Vladyslav Usenko
 Technical University of Munich

{nikolaus.demmel,d.schubert,c.sommer,cremers,vlad.usenko}@tum.de

A. Proofs and mathematical properties

A.1. Pseudo-Schur complement and SVD

First, we show the properties we use in Section 4.3.2 when proving that our proposed specialized QR decomposition is equivalent to using pseudo-Schur complement. This includes in particular the definition of the compact SVD for a rank-deficient matrix together with a definition of the Moore-Penrose inverse, as well as a result on subspaces of \mathbb{R}^n spanned by matrix columns.

Definition 1. Let $J \in \mathbb{R}^{n \times k}$ and $\text{rank}(J) = r \leq k$. The compact singular value decomposition (SVD) of J is of the form

$$J = U_1 D_1 V_1^\top, \quad (28)$$

where $U_1 \in \mathbb{R}^{n \times r}$, $D_1 \in \mathbb{R}^{r \times r}$, and $V_1 \in \mathbb{R}^{k \times r}$. D_1 is an invertible diagonal matrix with positive entries, $U_1^\top U_1 = V_1^\top V_1 = I_r$.

Thus, by definition of the compact SVD, the columns of U_1 span the column space of J . For the compact SVD of $J^\top J$, we get $V_1 D_1^2 V_1^\top$.

Definition 2. The Moore-Penrose inverse (also pseudo-inverse) of a matrix with compact SVD $U_1 D_1 V_1^\top$ is defined as

$$(U_1 D_1 V_1^\top)^+ = V_1 D_1^{-1} U_1^\top. \quad (29)$$

Thus, the pseudo-inverse of $(J^\top J)$ is given by

$$(J^\top J)^+ = V_1 D_1^{-2} V_1^\top. \quad (30)$$

Lemma 1. Let $Q, U \in \mathbb{R}^{n \times r}$, and let the columns of Q and U span the same r -dimensional subspace of \mathbb{R}^n . Further, let both Q and U have mutually orthogonal columns of norm 1, i.e., $Q^\top Q = U^\top U = I_r$. Then, the following holds:

$$QQ^\top = UU^\top. \quad (31)$$

Proof. Since the columns of Q and U span the same space, each column of Q can be written as a linear combination of the columns of U and vice versa. Thus, there is a matrix M

such that $Q^\top = MU^\top$ and $U^\top = M^{-1}Q^\top$. As $Q^\top Q = U^\top U = I_r$, $M = Q^\top U$ and $M^{-1} = U^\top Q = M^\top$. Thus, M is orthogonal, yielding

$$QQ^\top = UM^\top MU^\top = UU^\top. \quad (32)$$

□

A.2. Equivalence of pseudo-inverse and pseudo-Schur complement

In Sec. 4.3.3, we claim that under certain conditions, solving the full system using Moore-Penrose inverse is equivalent to using the generalized Schur complement followed by solving the reduced system with Moore-Penrose inverse. Moreover, a potential backsubstitution for the μ -variables can also be achieved using a Moore-Penrose inverse instead of an inverse:

$$\Delta x_{\mu, \text{red}} = H_{\mu\mu}^+ (b_\mu - H_{\mu\kappa} \Delta x_{\kappa, \text{red}}). \quad (33)$$

In the following, we will formalize and prove this statement.

Theorem 2. Let (26) hold, and let Δx_{tot} and Δx_{red} be defined as in Sec. 4.3.3. Then,

$$\Delta x_{\text{tot}} = \Delta x_{\text{red}}. \quad (34)$$

Proof. We start by noting that

$$\text{rank}(J_\mu) = \text{rank}(H_{\mu\mu}), \quad (35)$$

$$\text{rank} \begin{pmatrix} J_\kappa & J_u \\ H_{\kappa\kappa} & H_{\kappa u} \\ H_{u\kappa} & H_{uu} \end{pmatrix} =: r_{\kappa u}, \quad (36)$$

$$\text{rank}(J) = \text{rank}(H). \quad (37)$$

Thus, we can rewrite (26) as

$$\text{rank}(H) = \text{rank}(H_{\mu\mu}) + r_{\kappa u}, \quad (38)$$

and apply Lemma 2.3 from [14] with $A_{11} = H_{\mu\mu}$. This Lemma gives us a block-matrix expression for the pseudo-

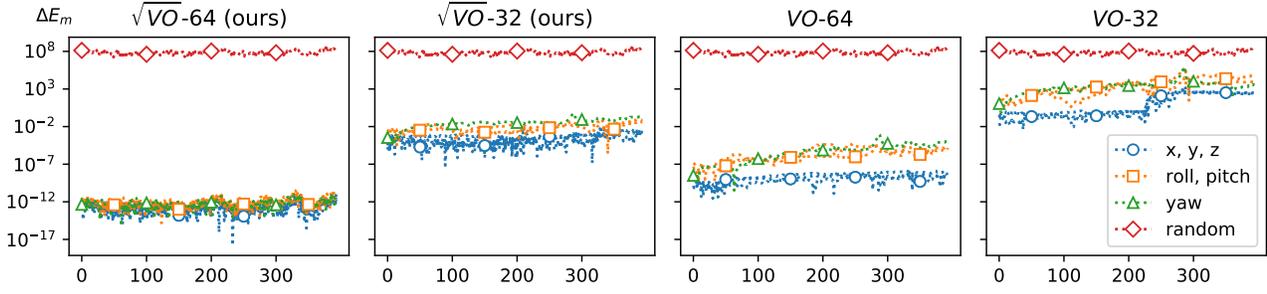


Figure 5. Where for visual-inertial odometry we expect 4 degrees of gauge freedom, for stereo visual odometry roll and pitch are not observable and we expect 6 degrees of gauge freedom. The plots show the marginalization prior cost change ΔE_m for VO on *kitti10* when perturbing the linearization point. For that, we consider perturbations by a global translation (in x , y , or z), by a global (linearized) rotation (roll, pitch, or yaw), or by a random unit norm vector. While our square root marginalization leads to a consistent prior with expected nullspaces for both single and double precision, in the conventional squared form accumulating error leads to inconsistency. Similar to the VIO case (compare Fig. 4), here for *VO-32* the prior over time erroneously appears to make the global pose observable, indicated by large cost change by perturbations in gauge direction. In particular, after around 200 keyframes there is a noticeable increase, which also coincides with worsened pose estimation (see Fig. 7).

inverse H^+ of H :

$$H^+ = \begin{pmatrix} A & -B^\top \\ -B & S^+ \end{pmatrix}, \quad (39)$$

$$S = \begin{pmatrix} \tilde{H} + H_{\kappa\kappa}^\mu & H_{\kappa u} \\ H_{u\kappa} & H_{uu} \end{pmatrix}, \quad (40)$$

$$B = S^+ \begin{pmatrix} H_{\kappa\mu} \\ 0 \end{pmatrix} H_{\mu\mu}^+, \quad (41)$$

$$A = H_{\mu\mu}^+ + H_{\mu\mu}^+ (H_{\mu\kappa} \ 0) B. \quad (42)$$

If we now compute $-H^+b$ and look at the κ - and u -components, we get

$$\begin{aligned} \begin{pmatrix} \Delta x_{\kappa, \text{tot}} \\ \Delta x_{u, \text{tot}} \end{pmatrix} &= B b_\mu - S^+ \begin{pmatrix} b_\kappa \\ b_u \end{pmatrix} \\ &= S^+ \begin{pmatrix} H_{\kappa\mu} H_{\mu\mu}^+ b_\mu - b_\kappa \\ -b_u \end{pmatrix} = -S^+ \begin{pmatrix} \tilde{b} + b_\kappa^\mu \\ b_u \end{pmatrix}, \end{aligned} \quad (43)$$

which is exactly the solution of (13), i.e.,

$$\begin{pmatrix} \Delta x_{\kappa, \text{tot}} \\ \Delta x_{u, \text{tot}} \end{pmatrix} = \begin{pmatrix} \Delta x_{\kappa, \text{red}} \\ \Delta x_{u, \text{red}} \end{pmatrix} \quad (44)$$

Similarly, from

$$\Delta x_{\mu, \text{tot}} = -A b_\mu + B^\top \begin{pmatrix} b_\kappa \\ b_u \end{pmatrix}, \quad (45)$$

after some steps, one obtains the back substitution formula (33)

$$\Delta x_{\mu, \text{tot}} = H_{\mu\mu}^+ (b_\mu - H_{\mu\kappa} \Delta x_{\kappa, \text{red}}) = \Delta x_{\mu, \text{red}}. \quad (46)$$

(44) and (46) together conclude the proof. \square

| | $\sqrt{\text{VO-64}}$ | $\sqrt{\text{VO-32}}$ | VO-64 | VO-32 |
|---------|-----------------------|-----------------------|-------------------|-------------------|
| kitti00 | 29.5 / 2.7 | 23.6 / 2.2 | 50.2 / 2.3 | x |
| kitti02 | 32.0 / 3.0 | 25.0 / 2.3 | 53.2 / 2.4 | x |
| kitti03 | 5.2 / 0.6 | 4.3 / 0.5 | 9.4 / 0.5 | 9.0 / 0.5 |
| kitti04 | 1.5 / 0.2 | 1.2 / 0.1 | 2.6 / 0.1 | 2.5 / 0.1 |
| kitti05 | 18.0 / 1.7 | 15.0 / 1.4 | 31.1 / 1.5 | x |
| kitti06 | 5.8 / 0.6 | 4.8 / 0.5 | 9.8 / 0.6 | 9.3 / 0.6 |
| kitti07 | 6.3 / 0.7 | 5.3 / 0.6 | 11.2 / 0.6 | 10.7 / 0.6 |
| kitti08 | 26.3 / 2.5 | 21.2 / 2.0 | 44.2 / 2.1 | x |
| kitti09 | 10.1 / 1.0 | 8.0 / 0.8 | 16.7 / 0.8 | x |
| kitti10 | 6.9 / 0.7 | 5.5 / 0.6 | 11.6 / 0.6 | 9.7 / 0.6 |

Table 5. Total runtime in seconds spent on “optimization / marginalization” in VO. *Optimization*: NS-projection for landmarks ($\sqrt{\text{VO-32}}$) is almost twice as fast as the baseline using SC (VO-64). *Marginalization*: conventional SC may be slightly faster, but this step only takes a small fraction of the overall runtime.

Note on square root of the κu -system While we have shown that $\tilde{H} = \tilde{R}^\top \tilde{R}$ and $\tilde{b} = \tilde{R}^\top \tilde{r}$, to complete the square root formulation, a square root of the system including u -variables as in (13) and (40) is given by

$$\tilde{R}_{\kappa u} = \begin{pmatrix} J_\kappa^\mu & J_u^\mu \\ \tilde{R} & 0 \end{pmatrix}, \quad \tilde{r}_{\kappa u} = \begin{pmatrix} r^\mu \\ \tilde{r} \end{pmatrix}. \quad (47)$$

B. Additional analysis of VO results

In this section we include additional results supporting the claims of the main paper. While these are for the same datasets, we expand upon some of the analysis that was omitted due to limited space. Specifically, we show runtimes, the ablation study, as well as the nullspace and eigenvalue analysis also for VO on the KITTI dataset. Qualita-

| opt. | proposed | | ablation study | | | | | |
|-------------|--------------|--------------|----------------|--------------|---------|-------------|--------------|-------|
| | NS+LDLT | SC+LDLT | NS+LDLT | SC+LDLT | NS+LDLT | SC+LDLT | | |
| marg. | NS+QR | NS+QR | SC+SC | SC+SC | NS+QR | SC+SC | | |
| precision | 64 | 32 | 64 | 32 | 64 | 32 | | |
| ATE [m] | 3.216 | 3.216 | 3.216 | 3.216 | 3.217 | 3.293 | 3.216 | 3.479 |
| real-time | 9.4x | 9.8x | 8.0x | 8.6x | 9.4x | 9.6x | 8.2x | 8.6x |
| t total [s] | 24.3 | 23.3 | 28.7 | 26.5 | 24.1 | 23.2 | 28.2 | 26.5 |
| t opt [s] | 14.2 | <i>11.3</i> | 24.2 | 22.6 | 14.1 | 11.0 | 24.2 | 22.2 |
| t marg [s] | 1.4 | <i>1.1</i> | 1.2 | 1.0 | 1.4 | 1.3 | 1.2 | 1.2 |

Table 6. Different combinations of optimization and marginalization techniques, and floating-point precision for \sqrt{VO} on KITTI. All variants store the marginalization prior in square root form (4). The shown metrics (ATE, runtime: total / optimization / marginalization) are averages over all sequences, and the real-time factor indicates how much faster the processing is compared to sequence duration. 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.

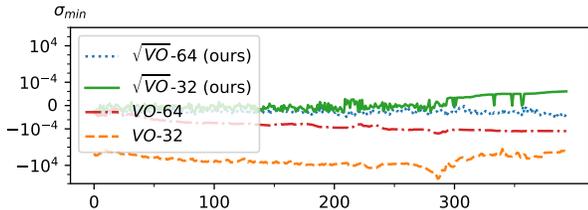


Figure 6. Smallest eigenvalue σ_{\min} of the marginalization prior Hessian H_m evolving over time for VO on *kitti10* (linear y-axis for $|\sigma_{\min}| < 10^{-8}$, logarithmic elsewhere). We expect values close to zero (positive semi-definite Hessian with gauge freedom). While the conventional (squared) formulation in single precision leads to negative eigenvalues with large magnitude (exceeding 10^8), accumulating error, and (ultimately) numerical failure, the proposed square root approach has σ_{\min} of bounded magnitude (less than 10^{-4}) and remains stable.

tively, these are similar to the VIO results from the main paper and thus we draw the same conclusions.

Tab. 5 shows runtimes for optimization and marginalization for VO (compare VIO results in Tab. 3). It can be seen that optimization takes a much bigger portion of total runtime than marginalization, that for the proposed single-precision solver \sqrt{VO} -32 it is around twice as fast as the competing baseline VO -64, and that the square root formulation benefits more in terms of runtime from switching from double to single precision.

Tab. 6 shows the same ablation study as Tab. 4, but for VO instead of VIO. Note that for KITTI, the twofold improvement in optimization runtime is not fully reflected in an improvement of total runtime. The reason is that here the optical flow, which is computed in a single parallel thread, becomes the bottleneck. However, the improved optimiza-

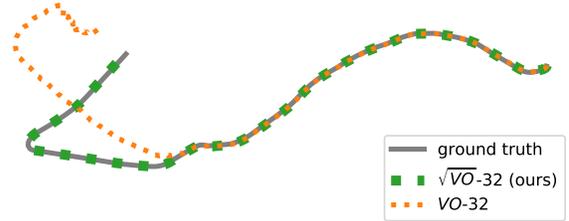


Figure 7. Estimated visual odometry trajectories on the *kitti10* sequence. The conventional baseline VO -64 works well with double-precision floats, but fails in single precision (VO -32). In contrast, the proposed square root estimator \sqrt{VO} -32 even in single precision retains the same accuracy.

tion runtime still means the required compute power is reduced. Overall, also for VO we conclude that only the combination of all proposed improvements leads to best accuracy and runtime.

The analysis of numerical properties of the marginalization prior Hessian of VO on *kitti10* reveals similar behaviour to VIO (see Sec. 5.2). For the squared formulation in single precision the marginalization prior becomes numerically indefinite (Fig. 6, compare VIO results in Fig. 1 bottom) and gauge freedom vanishes (Fig. 5, compare VIO results in Fig. 4). While initially the pose estimation works fine, at some point the accumulating error leads to bad state estimates and ultimately numerical failure (Fig. 7, compare Fig. 1 top). In contrast, the proposed \sqrt{VO} has the same accuracy in both single and double precision, at a significantly reduced computational cost.

C. Notes on memory overhead

The main memory requirement of our optimization and marginalization comes from the dense landmark blocks, where we perform QR on the Jacobians in-place to marginalize landmarks. [6] reports around twice the memory use compared to SC for sparse BA problems and mentions memory to be the limiting factor for large dense problems. However, for us the number of keyframes and number of observations per landmark are bounded in the sliding window and thus memory use is not a major concern.

For example, for VIO on Euroc MH01 we have at most 4033 observations across all landmarks, and at most 7 keyframes (3 with IMU, state size 15, and 4 pose-only, state size 6), so the Jacobians have in total 8066 rows and 73 columns (3+1 extra for landmark+residual), giving an approximate upper bound of 2.4MB with 32bit floats. Measuring the actual difference in peak memory between the single and double precision variants reveals 1.3MB for the

square root solver and 0.9MB for the SC solver, while the vast majority of process peak memory at around 300MB is spent in other parts of the (not memory-optimized) system (e.g. cached image queue, logging, etc...).

A memory-conscious implementation could in fact reduce the required landmark-block memory by doing a one-pass over landmarks that linearizes, marginalizes and accumulates the RCS Hessian using scratch memory. Only 3 rows per landmark for back-substitution would need to be stored.