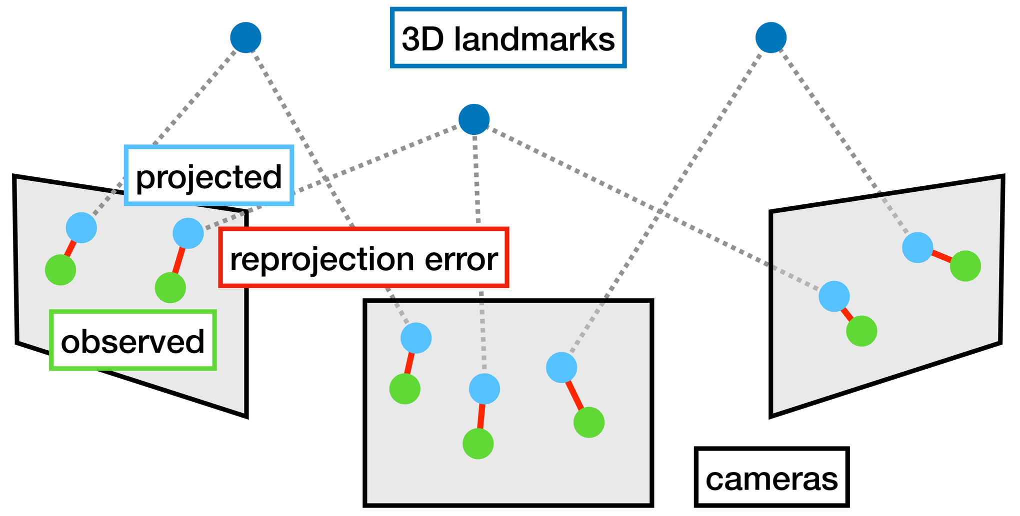
Square Root Bundle Adjustment for Large-Scale Reconstruction

Nikolaus Demmel, Christiane Sommer, Daniel Cremers and Vladyslav Usenko

Computer Vision Group, Technical University of Munich

The bundle adjustment (BA) problem

Bundle adjustment is the joint refinement of camera poses and 3D landmarks. It is essential for many vision applications such as Structure from Motion, 3D reconstruction and SLAM. Large scale means thousands of cameras and millions of points.



Reprojection error of 3D landmark X_j observed at pixel position u_{ij} in frame i with camera pose (R_i, t_i) and intrinsics c_i:

$$r_{ij} = u_{ij} - \pi(R_i X_j + t_i; c_i) .$$

Non-linear least squares energy for stacked variables $x_p = \{R_i, t_i, c_i\}_i$ and $x_l = \{X_j\}_j$:

$$E(x_p, x_l) = \sum_{i,j} \|r_{ij}\|^2 = \|r(x_p, x_l)\|^2$$

QR decomposition

Let $A \in \mathbb{R}^{m \times n}$ have full rank $\operatorname{rank}(A) = n \leq m$. The QR decomposition of A is

$$A = QR = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = (Q_1 \; Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1 .$$

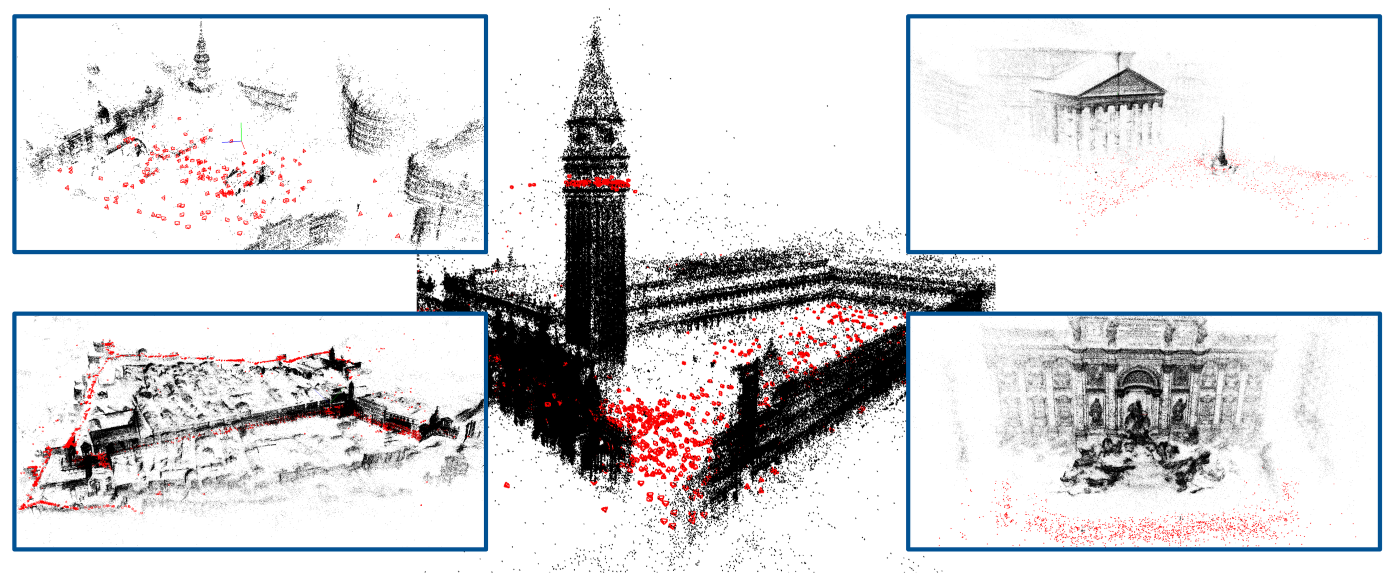
The columns of Q_2 form the left nullspace of A: $Q_2^\top A = 0$.

Acknowledgements

This work was supported by the Munich Center for Machine Learning, by the ERC Advanced Grant *SIMULACRON* and by the DFG project CR 250/20-1 *Splitting Methods for 3D Reconstruction and SLAM.*

Summary

We propose **nullspace marginalization** to reduce the system size in bundle adjustment problems and prove that it is **algebraically equivalent** to Schur complement. For use in a Levenberg-Marquardt solver with Preconditioned Conjugate Gradient, we present an efficient **implementation strategy** that is well **parallelizable** and allows computation in **single precision** with little loss of solution quality. We perform extensive evaluation on the real-world BAL datasets, demonstrating **significantly reduced runtime** compared to Ceres Solver and our own Schur complement-based implementation.



Landmark marginalization

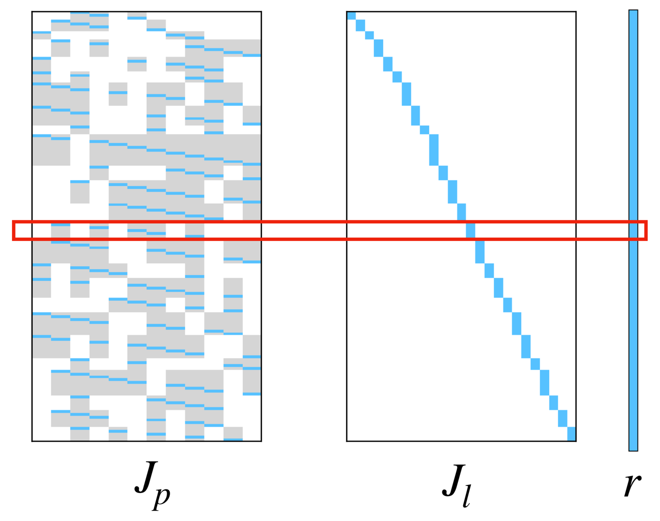
Linear system:

$$E_{\text{lin}}(\Delta x_p, \Delta x_l) = \left\| r + (J_p \; J_l) \begin{pmatrix} \Delta x_p \\ \Delta x_l \end{pmatrix} \right\|^2$$

Nullspace marginalization:

$$\min_{\Delta x_p} \|Q_2^\top r + Q_2^\top J_p \Delta x_p\|^2$$

where $J_l = QR$

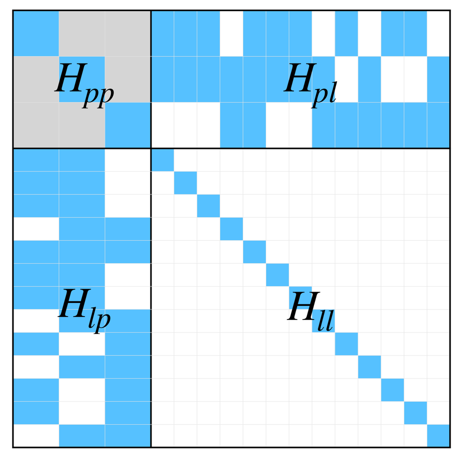


Normal equations:

$$\begin{pmatrix} H_{pp} & H_{pl} \\ H_{lp} & H_{ll} \end{pmatrix} \begin{pmatrix} -\Delta x_p \\ -\Delta x_l \end{pmatrix} = \begin{pmatrix} b_p \\ b_l \end{pmatrix}$$

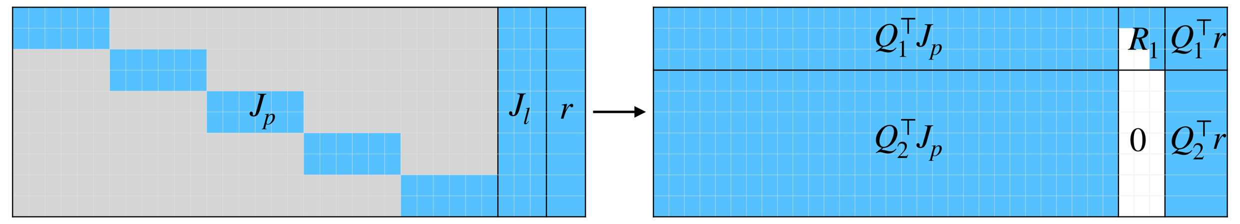
Schur complement (RCS):

$$\tilde{H}_{pp}(-\Delta x_p) = \tilde{b}_p$$



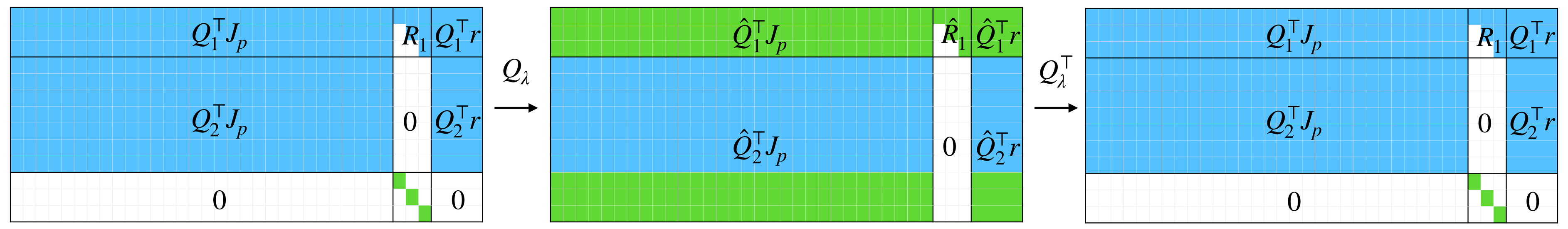
Implementation Strategy

Marginalization in Landmark Blocks:



- dense storage per landmark block
- compute QR factorization of J_l
- apply Givens rotations in-place
- all steps parallelizable over landmarks
- apply damping with 6 Givens rotations

Damping in Landmark Blocks:



PCG with Landmark Blocks:
Compute once:

$$\tilde{b}_p = (Q_2^\top J_p)^\top (Q_2^\top r) .$$

Compute in every CG iteration:

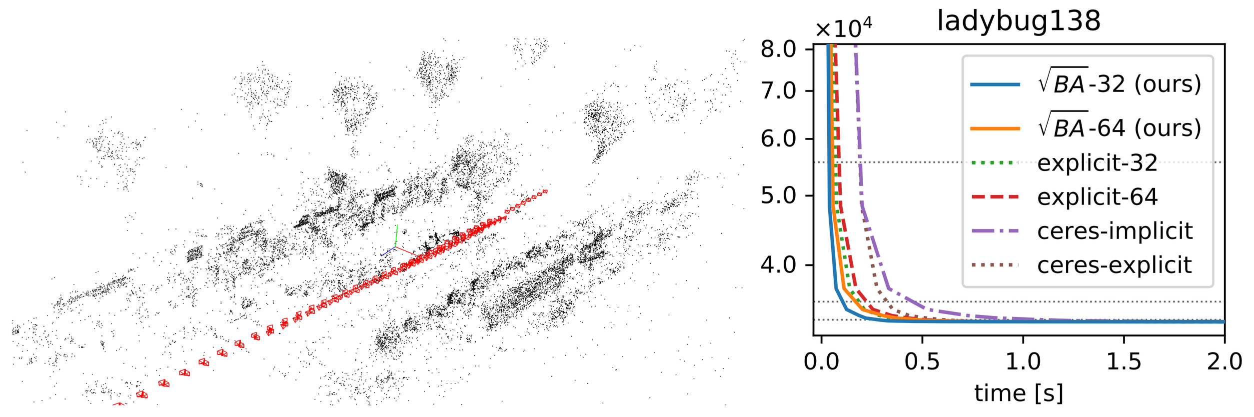
$$\tilde{H}_{pp}\, v = (Q_2^\top J_p)^\top (Q_2^\top J_p\, v) .$$

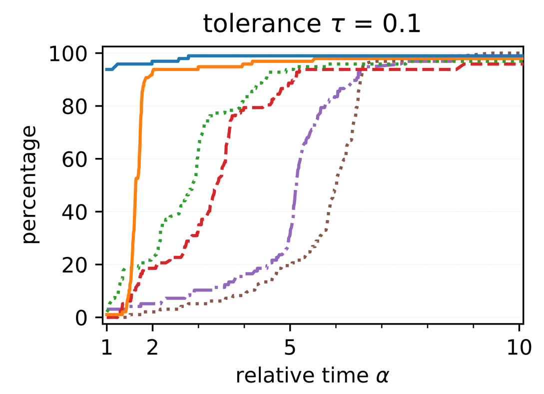
PCG with Damping:

$$\tilde{H}_{pp}\, v = (\hat{Q}_2^\top J_p)^\top (\hat{Q}_2^\top J_p\, v) + \lambda D_p^2\, v .$$

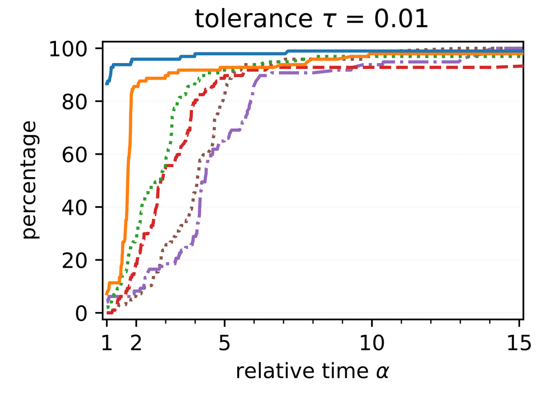
Results: Convergence Plots and Performance Profiles

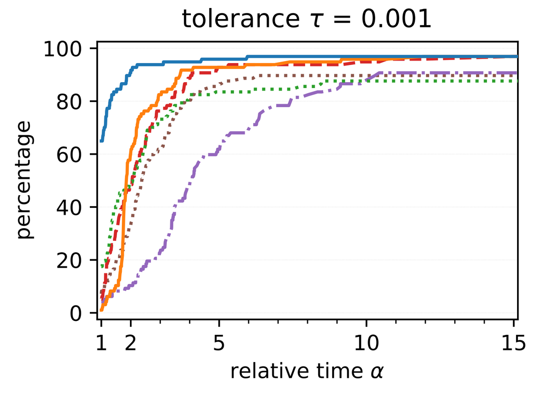
Rendered optimized landmark point cloud and **convergence plot** for ladybug138. All solvers reach a similar cost, but the proposed square root BA is the fastest.





Performance profiles show percentage of all 97 BAL problems solved to a given accuracy tolerance τ with increasing relative runtime α. A curve more to the left and top indicates better runtime and accuracy.





Code & Contact

Code is available open-source:
`https://go.vision.in.tum.de/rootba`

Contact: Nikolaus Demmel
`nikolaus.demmel@tum.de`