Supplementary Material for Revisiting Rolling Shutter Bundle Adjustment: Toward Accurate and Fast Solution

Bangyan Liao[1,2,*] Delin Qu[1,3,*] Yifei Xue[4] Huiqing Zhang[1] Yizhen Lao[1,†]
[1]College of Computer Science and Electronic Engineering, Hunan University
[2]College of Electrical and Information Engineering, Hunan University
[3]Shanghai AI Laboratory
[4]Jiangxi Provincial Natural Resources Cause Development Center

A. Proof of Reprojection Error Covariance

In this section, we perform a detailed proof of reprojection error covariance. Firstly, we decompose a normalized image measurement point $\mathbf{q}_i^j = \begin{bmatrix} c & r \end{bmatrix}^\top$ into a perfect normalized image measurement point $\tilde{\mathbf{q}}_i^j = \begin{bmatrix} \tilde{c} & \tilde{r} \end{bmatrix}^\top$ and a normalized image Gaussian measurement noise $\begin{bmatrix} n_c & n_r \end{bmatrix}^\top$:

$$\begin{bmatrix} c & r \end{bmatrix}^\top = \begin{bmatrix} \tilde{c} & \tilde{r} \end{bmatrix}^\top + \begin{bmatrix} n_c & n_r \end{bmatrix}^\top, \tag{1}$$

with,

$$\begin{bmatrix} n_c & n_r \end{bmatrix}^\top \sim \mathcal{N}(0, \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top), \tag{2}$$

$$\mathbf{W} = \begin{bmatrix} 1/f_x & 0 \\ 0 & 1/f_y \end{bmatrix}, \tag{3}$$

where $\mathbf{\Sigma}$ is the prior Gaussian measurement noise and f_x f_y are the x-axis and y-axis focal length respectively. Then we can substitute Eq. (1) to the normalized measurement based reprojection cost function as:

$$\begin{aligned} \mathbf{e}_i^j &= \begin{bmatrix} c_i^j & r_i^j \end{bmatrix}^\top - \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j)) \\ &= \begin{bmatrix} c_i^j & r_i^j \end{bmatrix}^\top - \Pi(\mathbf{R}^j(\tilde{r}_i^j + n_r)\mathbf{P}_i + \mathbf{t}^j(\tilde{r}_i^j + n_r)), \end{aligned} \tag{4}$$

where $\Pi(\mathbf{R}^j(\tilde{r}_i^j + n_r)\mathbf{P}_i + \mathbf{t}^j(\tilde{r}_i^j + n_r))$ can be linearized using the Taylor first order approximation:

$$\begin{aligned} &\Pi(\mathbf{R}^j(\tilde{r}_i^j + n_r)\mathbf{P}_i + \mathbf{t}^j(\tilde{r}_i^j + n_r)) \\ &\approx \begin{bmatrix} \tilde{c}_i^j \\ \tilde{r}_i^j \end{bmatrix} + \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial n_r} n_r. \end{aligned} \tag{5}$$

Then by substituting Eq. (5) into Eq. (4), we have

$$\mathbf{e}_i^j = \begin{bmatrix} n_c \\ n_r \end{bmatrix} - \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial n_r} n_r = \mathbf{C}_i^j \begin{bmatrix} n_c \\ n_r \end{bmatrix}. \tag{6}$$

By applying the chain rule of derivation, we can get the analytical formulation of matrix $\mathbf{C}_i^j$.

$$\frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial n_r} = \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial \mathbf{P}^{cj}_i} \frac{\partial \mathbf{P}^{cj}_i}{\partial n_r}, \tag{7}$$

with,

$$\frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial \mathbf{P}^{cj}_i} = \begin{bmatrix} \frac{1}{Z^{cj}_i} & 0 \\ 0 & \frac{1}{Z^{cj}_i} \\ \frac{-X^{cj}_i}{Z^{cj\,2}_i} & \frac{-Y^{cj}_i}{Z^{cj\,2}_i} \end{bmatrix}^\top, \tag{8}$$

$$\frac{\partial \mathbf{P}^{cj}_i}{\partial n_r} = [\boldsymbol{\omega}^j]_\times \mathbf{R}^j \mathbf{P}_i + \mathbf{d}^j. \tag{9}$$

By substituting Eq. (7) into Eq. (6), we can get analytical formulation:

$$\mathbf{C}_i^j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{Z^{cj}_i} & 0 \\ 0 & \frac{1}{Z^{cj}_i} \\ \frac{-X^{cj}_i}{Z^{cj\,2}_i} & \frac{-Y^{cj}_i}{Z^{cj\,2}_i} \end{bmatrix}^\top ([\boldsymbol{\omega}^j]_\times \mathbf{R}^j \mathbf{P}_i + \mathbf{d}^j) \begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top, \tag{10}$$

where $\mathbf{P}^{cj}_i = \begin{bmatrix} X^{cj}_i & Y^{cj}_i & Z^{cj}_i \end{bmatrix}^\top = \mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j)$ is the world point $\mathbf{P}_i$ in camera j coordinates. Combining Eq. (2) with Eq. (6), we prove that $\mathbf{e}_i^j$ follows a weighted Gaussian distribution:

$$\mathbf{e}_i^j \sim \mathcal{N}(0, \mathbf{C}_i^j \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top \mathbf{C}_i^{j\,\top}). \tag{11}$$

* Authors contributed equally
† Corresponding author: yizhenlao@hnu.edu.cn
Project page: https://delinqu.github.io/NW-RSBA

B. Analytical Jacobian matrix Derivation

In this section, we provide a detailed derivation of the analytical Jacobian matrix used in our proposed *NW-RSBA* solution.

B.1. Jacobian Matrix Parameterization

To derive the analytical Jacobian matrix of Eq. (15), we use $\boldsymbol{\xi}_i^j \in \mathbf{so}(3)$ to parametrize $\mathbf{R}_i^j \in \mathbf{SO}(3)$. These two representations can be transformed to each other by Rodrigues formulation $\mathbf{R}_i^j = \mathbf{Exp}(\boldsymbol{\xi}_i^j)$ and $\boldsymbol{\xi}_i^j = \mathbf{Log}(\mathbf{R}_i^j)$, which are defined as:

$$\begin{aligned}\mathbf{R} &= \mathbf{Exp}(\boldsymbol{\xi}) \\ &= \mathbf{I} + \frac{sin(\|\boldsymbol{\xi}\|)}{\|\boldsymbol{\xi}\|}\boldsymbol{\xi}^{\wedge} + \frac{1 - cos(\|\boldsymbol{\xi}\|)}{\|\boldsymbol{\xi}\|^2}(\boldsymbol{\xi}^{\wedge})^2.\end{aligned} \tag{12}$$

$$\begin{aligned}\boldsymbol{\xi} &= \mathbf{Log}(\mathbf{R}) \\ &= \frac{\boldsymbol{\theta}}{2sin(\boldsymbol{\theta})}(\mathbf{R} - \mathbf{R}^{\top})^{\vee},\end{aligned} \tag{13}$$

with,

$$\boldsymbol{\theta} = arccos((tr(\mathbf{R}) - 1)/2), \tag{14}$$

where $\wedge$ is the skew-symmetric operator that can transform a vector to the corresponding skew-symmetric matrix. Besides, $\vee$ is the inverse operator.

B.2. Partial Derivative of Reprojection error

Recall the normalized weighted error term, which is defined as:

$$\hat{\mathbf{e}}_i^j = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{W}^{-1}{\mathbf{C}_i^j}^{-1}\mathbf{e}_i^j. \tag{15}$$

Then we can get five atomic partial derivatives of $\partial\hat{\mathbf{e}}_i^j$ over $\partial\mathbf{P}_i$, $\partial\boldsymbol{\xi}^j$, $\partial\mathbf{t}^j$, $\partial\boldsymbol{\omega}^j$ and $\partial\mathbf{d}^j$ as:

$$\frac{\partial\hat{\mathbf{e}}_i^j}{\partial\mathbf{P}_i} = {\boldsymbol{\Sigma}_n}^{-\frac{1}{2}}\mathbf{W}^{-1}({\mathbf{C}_i^j}^{-1}\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j}\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{P}_i} + \begin{bmatrix}{\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\mathbf{P}_i} \\ {\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(2)^{-\top}}{\partial\mathbf{P}_i}\end{bmatrix}), \tag{16}$$

$$\frac{\partial\hat{\mathbf{e}}_i^j}{\partial\boldsymbol{\xi}^j} = {\boldsymbol{\Sigma}_n}^{-\frac{1}{2}}\mathbf{W}^{-1}({\mathbf{C}_i^j}^{-1}\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j}\frac{\partial{\mathbf{P}^c}_i^j}{\partial\boldsymbol{\xi}^j} + \begin{bmatrix}{\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\boldsymbol{\xi}^j} \\ {\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(2)^{-\top}}{\partial\boldsymbol{\xi}^j}\end{bmatrix}), \tag{17}$$

$$\frac{\partial\hat{\mathbf{e}}_i^j}{\partial\mathbf{t}^j} = {\boldsymbol{\Sigma}_n}^{-\frac{1}{2}}\mathbf{W}^{-1}({\mathbf{C}_i^j}^{-1}\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j}\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{t}^j} + \begin{bmatrix}{\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\mathbf{t}^j} \\ {\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(2)^{-\top}}{\partial\mathbf{t}^j}\end{bmatrix}), \tag{18}$$

$$\frac{\partial\hat{\mathbf{e}}_i^j}{\partial\boldsymbol{\omega}^j} = {\boldsymbol{\Sigma}_n}^{-\frac{1}{2}}\mathbf{W}^{-1}({\mathbf{C}_i^j}^{-1}\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j}\frac{\partial{\mathbf{P}^c}_i^j}{\partial\boldsymbol{\omega}^j} + \begin{bmatrix}{\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\boldsymbol{\omega}^j} \\ {\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(2)^{-\top}}{\partial\boldsymbol{\omega}^j}\end{bmatrix}), \tag{19}$$

$$\frac{\partial\hat{\mathbf{e}}_i^j}{\partial\mathbf{d}^j} = {\boldsymbol{\Sigma}_n}^{-\frac{1}{2}}\mathbf{W}^{-1}({\mathbf{C}_i^j}^{-1}\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j}\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{d}^j} + \begin{bmatrix}{\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\mathbf{d}^j} \\ {\mathbf{e}_i^j}^{\top}\frac{\partial\mathbf{C}_i^j(2)^{-\top}}{\partial\mathbf{d}^j}\end{bmatrix}), \tag{20}$$

with,

$$\frac{\partial\mathbf{e}_i^j}{\partial{\mathbf{P}^c}_i^j} = -\begin{bmatrix}\frac{1}{{Z^c}_i^j} & 0 & -\frac{{X^c}_i^j}{({Z^c}_i^j)^2} \\ 0 & \frac{1}{{Z^c}_i^j} & -\frac{{Y^c}_i^j}{({Z^c}_i^j)^2}\end{bmatrix}, \tag{21}$$

$$\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{P}_i} = (\mathbf{I} + [\boldsymbol{\omega}^j]_{\times}r_i^j)\mathbf{R}^j, \tag{22}$$

$$\frac{\partial{\mathbf{P}^c}_i^j}{\partial\boldsymbol{\xi}^j} = -(\mathbf{I} + [\boldsymbol{\omega}^j]_{\times}r_i^j)[\mathbf{R}^j\mathbf{P}_i]_{\times}, \tag{23}$$

$$\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{t}^j} = [\mathbf{I}]_{3\times3}, \tag{24}$$

$$\frac{\partial{\mathbf{P}^c}_i^j}{\partial\boldsymbol{\omega}^j} = -r_i^j[\mathbf{R}^j\mathbf{P}_i]_{\times}, \tag{25}$$

$$\frac{\partial{\mathbf{P}^c}_i^j}{\partial\mathbf{d}^j} = r_i^j[\mathbf{I}]_{3\times3}, \tag{26}$$

where $\mathbf{C}_i^j(1)$ and $\mathbf{C}_i^j(2)$ represents the first and second row of matrix $\mathbf{C}_i^j$ respectively.

We further need to derive the partial derivatives of $\partial\mathbf{C}_i^j(.)^{-\top}$ over $\partial\mathbf{P}_i$, $\partial\boldsymbol{\xi}^j$, $\partial\mathbf{t}^j$, $\partial\boldsymbol{\omega}^j$ and $\partial\mathbf{d}^j$ in Eq. (16 - 20). Recall the $\mathbf{C}_i^j$ and $\mathbf{W}$ definition in Eq. (10) and Eq. (3). For convenience, we define the following two intermediate variables:

$$\boldsymbol{\gamma}_i^j = \begin{bmatrix}\frac{1}{{Z^c}_i^j} & 0 \\ 0 & \frac{1}{{Z^c}_i^j} \\ \frac{-{X^c}_i^j}{{Z^c}_i^{j\,2}} & \frac{-{Y^c}_i^j}{{Z^c}_i^{j\,2}}\end{bmatrix}^{\top}, \tag{27}$$

$$\boldsymbol{\delta}_i^j = [\boldsymbol{\omega}^j]_{\times}\mathbf{R}^j\mathbf{P}_i + \mathbf{d}^j. \tag{28}$$

Then we can rewrite Eq. (10) as:

$$\mathbf{C}_i^j = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} - \boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j\begin{bmatrix}0 & 1\end{bmatrix} = \begin{bmatrix}1 & -\boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j \\ 0 & 1 - \boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j\end{bmatrix}, \tag{29}$$

and its inverse formulation as:

$${\mathbf{C}_i^j}^{-1} = \begin{bmatrix}1 & \frac{\boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j}{1-\boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j} \\ 0 & \frac{1}{1-\boldsymbol{\gamma}_i^j\boldsymbol{\delta}_i^j}\end{bmatrix}. \tag{30}$$

Then we can derive the partial derivative as:

$$\frac{\partial\mathbf{C}_i^j(1)^{-\top}}{\partial\mathbf{P}_i} = \begin{bmatrix}[0]_{1\times3} \\ \frac{(1-\boldsymbol{\beta})\frac{\partial\boldsymbol{\alpha}_i^j}{\partial\mathbf{P}_i} + \boldsymbol{\alpha}\frac{\partial\boldsymbol{\beta}_i^j}{\partial\mathbf{P}_i}}{(1-\boldsymbol{\beta}_i^j)^2}\end{bmatrix}, \tag{31}$$

$$\frac{\partial \mathbf{C}_i^j(2)}{\partial \mathbf{P}_i}^{-\top} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{\frac{\partial \boldsymbol{\beta}_i^j}{\partial \mathbf{P}_i}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{32}$$

$$\frac{\partial \mathbf{C}_i^j(1)^{-\top}}{\partial \boldsymbol{\xi}^j} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{(1-\boldsymbol{\beta}_i^j)\frac{\partial \boldsymbol{\alpha}_i^j}{\partial \boldsymbol{\xi}^j} + \boldsymbol{\alpha}_i^j \frac{\partial \boldsymbol{\beta}_i^j}{\partial \boldsymbol{\xi}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{33}$$

$$\frac{\partial \mathbf{C}_i^j(2)}{\partial \boldsymbol{\xi}^j}^{-\top} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{\frac{\partial \boldsymbol{\beta}_i^j}{\partial \boldsymbol{\xi}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{34}$$

$$\frac{\partial \mathbf{C}_i^j(1)^{-\top}}{\partial \mathbf{t}^j} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{(1-\boldsymbol{\beta}_i^j)\frac{\partial \boldsymbol{\alpha}_i^j}{\partial \mathbf{t}^j} + \boldsymbol{\alpha}_i^j \frac{\partial \boldsymbol{\beta}_i^j}{\partial \mathbf{t}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{35}$$

$$\frac{\partial \mathbf{C}_i^j(2)}{\partial \mathbf{t}^j}^{-\top} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{\frac{\partial \boldsymbol{\beta}_i^j}{\partial \mathbf{t}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{36}$$

$$\frac{\partial \mathbf{C}_i^j(1)^{-\top}}{\partial \boldsymbol{\omega}^j} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{(1-\boldsymbol{\beta}_i^j)\frac{\partial \boldsymbol{\alpha}_i^j}{\partial \boldsymbol{\omega}^j} + \boldsymbol{\alpha}_i^j \frac{\partial \boldsymbol{\beta}_i^j}{\partial \boldsymbol{\omega}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{37}$$

$$\frac{\partial \mathbf{C}_i^j(2)}{\partial \boldsymbol{\omega}^j}^{-\top} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{\frac{\partial \boldsymbol{\beta}_i^j}{\partial \boldsymbol{\omega}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{38}$$

$$\frac{\partial \mathbf{C}_i^j(1)^{-\top}}{\partial \mathbf{d}^j} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{(1-\boldsymbol{\beta}_i^j)\frac{\partial \boldsymbol{\alpha}_i^j}{\partial \mathbf{d}^j} + \boldsymbol{\alpha}_i^j \frac{\partial \boldsymbol{\beta}_i^j}{\partial \mathbf{d}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{39}$$

$$\frac{\partial \mathbf{C}_i^j(2)}{\partial \mathbf{d}^j}^{-\top} = \begin{bmatrix} [0]_{1\times 3} \\ \frac{\frac{\partial \boldsymbol{\beta}_i^j}{\partial \mathbf{d}^j}}{(1-\boldsymbol{\beta}_i^j)^2} \end{bmatrix}, \tag{40}$$

where $\mathbf{C}(1)$ and $\mathbf{C}(2)$ are the first and second row of $\mathbf{C}$ respectively, and two intermediate variables $\boldsymbol{\alpha}_i^j$ $\boldsymbol{\beta}_i^j$ are the first and second row of $\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j$ respectively

$$\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j = \begin{bmatrix} \boldsymbol{\alpha}_i^j \\ \boldsymbol{\beta}_i^j \end{bmatrix}. \tag{41}$$

Finally we have to derive the partial derivative of $\partial \boldsymbol{\alpha}_i^j$ and $\partial \boldsymbol{\beta}_i^j$ over $\partial \mathbf{P}_i$, $\partial \boldsymbol{\xi}^j$, $\partial \mathbf{t}^j$, $\partial \boldsymbol{\omega}^j$ and $\partial \mathbf{d}^j$ in Eq. (31 - 40):

$$\frac{\partial (\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j)}{\partial \mathbf{P}_i} = \boldsymbol{\gamma}_i^j \frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{P}_i} + \begin{bmatrix} \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{P}_i} \\ \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{P}_i} \end{bmatrix}, \tag{42}$$

$$\frac{\partial (\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j)}{\partial \boldsymbol{\xi}^j} = \gamma_i^j \frac{\partial \boldsymbol{\delta}_i^j}{\partial \boldsymbol{\xi}^j} + \begin{bmatrix} \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \boldsymbol{\xi}^j} \\ \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \boldsymbol{\xi}^j} \end{bmatrix}, \tag{43}$$

$$\frac{\partial (\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j)}{\partial \mathbf{t}^j} = \boldsymbol{\gamma}_i^j \frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{t}^j} + \begin{bmatrix} \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{t}^j} \\ \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{t}^j} \end{bmatrix}, \tag{44}$$

$$\frac{\partial (\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j)}{\partial \boldsymbol{\omega}^j} = \boldsymbol{\gamma}_i^j \frac{\partial \boldsymbol{\delta}_i^j}{\partial \boldsymbol{\omega}^j} + \begin{bmatrix} \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \boldsymbol{\omega}^j} \\ \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \boldsymbol{\omega}^j} \end{bmatrix}, \tag{45}$$

$$\frac{\partial (\boldsymbol{\gamma}_i^j \boldsymbol{\delta}_i^j)}{\partial \mathbf{d}^j} = \boldsymbol{\gamma}_i^j \frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{d}^j} + \begin{bmatrix} \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{d}^j} \\ \boldsymbol{\delta}_i^{j\top} \frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} \frac{\partial \mathbf{P}_i^{cj}}{\partial \mathbf{d}^j} \end{bmatrix}, \tag{46}$$

with,

$$\frac{\partial \boldsymbol{\gamma}_i^j(1)^\top}{\partial \mathbf{P}_i^{cj}} = \begin{bmatrix} 0 & 0 & -\frac{1}{{Z_i^{cj}}^2} \\ 0 & 0 & 0 \\ -\frac{1}{{Z_i^{cj}}^2} & 0 & \frac{2X_i^{cj}}{{Z_i^{cj}}^3} \end{bmatrix}, \tag{47}$$

$$\frac{\partial \boldsymbol{\gamma}_i^j(2)^\top}{\partial \mathbf{P}_i^{cj}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{{Z_i^{cj}}^2} \\ 0 & -\frac{1}{{Z_i^{cj}}^2} & \frac{2Y_i^{cj}}{{Z_i^{cj}}^3} \end{bmatrix}, \tag{48}$$

$$\frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{P}_i} = [\boldsymbol{\omega}^j]_\times \mathbf{R}^j, \tag{49}$$

$$\frac{\partial \boldsymbol{\delta}_i^j}{\partial \boldsymbol{\xi}^j} = [\boldsymbol{\omega}^j]_\times [\mathbf{R}^j \mathbf{P}_i]_\times, \tag{50}$$

$$\frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{t}^j} = [0]_{3\times 3}, \tag{51}$$

$$\frac{\partial \boldsymbol{\delta}_i^j}{\partial \boldsymbol{\omega}^j} = -[\mathbf{R}^j \mathbf{P}_i]_\times, \tag{52}$$

$$\frac{\partial \boldsymbol{\delta}_i^j}{\partial \mathbf{d}^j} = [\mathbf{I}]_{3\times 3}. \tag{53}$$

C. The proof of degeneracy resilience ability

As proved in [1], under the planar degeneracy configuration, the y-component of the reprojection error will reduce to zero. To say it in another way, the noise perturbation along the y-component of the observation will not be reflected in the y-component of the reprojection error (remains at zero). The reprojection error covariance matrix must have a zero variance in the y-coordinate of its values according to the definition of covariance. We prove this theoretically in the following.

In correspondence with notations in the manuscript, we first define $\mathbf{P}_i^{gj} = [X_i^{gj}\ Y_i^{gj}\ Z_i^{gj}]^\top = \mathbf{R}_0^j\mathbf{P}_i + \mathbf{t}_0^j$ and it can be related with $\mathbf{P}_i^{cj}$ as $[\boldsymbol{\omega}^j]_\times \mathbf{R}_0^j\mathbf{P}_i + \mathbf{d}^j = (\mathbf{P}_i^{cj} - \mathbf{P}_i^{gj}) \,/\, v_i^j$. Then we rewrite the Eq. (10):

$$\begin{aligned}\mathbf{C}_i^j &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}\frac{1}{Z_i^{cj}} & 0\\ 0 & \frac{1}{Z_i^{cj}}\\ \frac{-X_i^{cj}}{Z_i^{cj^2}} & \frac{-Y_i^{cj}}{Z_i^{cj^2}}\end{bmatrix}^\top \frac{\mathbf{P}_i^{cj} - \mathbf{P}_i^{gj}}{v_i^j} \begin{bmatrix}0\\1\end{bmatrix}^\top \\ &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}\frac{Z_i^{gj}X_i^{cj} - Z_i^{cj}X_i^{gj}}{v_i^j Z_i^{cj^2}}\\ \frac{Z_i^{gj}Y_i^{cj} - Z_i^{cj}Y_i^{gj}}{v_i^j Z_i^{cj^2}}\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix}^\top.\end{aligned} \tag{54}$$

Under the degeneracy configuration, the observed point will project to the plane $y = 0$ in the camera coordinate. We then substitute the degeneracy condition $Y_i^{gj} = 0, Z_i^{gj} = Z_i^{cj}, v_i^j = Y_i^{cj}/Z_i^{cj}$ into Eq.(54). It can be verified that the lower right component of $\mathbf{C}_i^j$ reduces to zero, which means that the y-coordinate variance in the reprojection error covariance matrix will reduce to zero.

Based on the explicitly modeled reprojection error covariance, we can decompose its inverse form and then reweight the reprojection error Eq.(11-13), which will result an isotropy covariance (Fig. 3). The corresponding weight will rapidly approach infinite during the degeneracy process since the y-coordinate variance gradually reduces to zero. As a result, the reweighted reprojection error will grow exponentially, which will prevent the continuation of the degeneracy. The error of *NM-RSBA* decreases gradually during the degeneracy process, while *NW-RSBA* grows exponentially and converges around the ground truth.

D. The equivalent between Normalized DC-RSBA and NW-RSBA

In this section, we provide an equivalent proof and illustrate the deep connection between the Normalized *DC-RSBA* and proposed *NW-RSBA* method.

D.1. Pre-definition

Recall the Eq. (6), we define a new vector $\boldsymbol{\chi}_i^j$ for convenience:

$$\begin{aligned}\boldsymbol{\chi}_i^j &= \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial n_r} \\ &= \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial r_i^j},\end{aligned} \tag{55}$$

then we can reformulate Eq. (10) as:

$$\begin{aligned}\mathbf{C}_i^j &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial n_r} \begin{bmatrix}0 & 1\end{bmatrix} \\ &= \begin{bmatrix}1 & -\boldsymbol{\chi}_i^j(1)\\ 0 & 1 - \boldsymbol{\chi}_i^j(2)\end{bmatrix},\end{aligned} \tag{56}$$

where $\boldsymbol{\chi}_i^j(1)$ $\boldsymbol{\chi}_i^j(2)$ are the first and second row of $\boldsymbol{\chi}_i^j$, and its inverse formulation is defined as:

$$\mathbf{C}_i^{j\,-1} = \begin{bmatrix}1 & \frac{\boldsymbol{\chi}_i^j(1)}{1-\boldsymbol{\chi}_i^j(2)}\\ 0 & \frac{1}{1-\boldsymbol{\chi}_i^j(2)}\end{bmatrix}. \tag{57}$$

Then we can define a new rectified image coordinate vector $\begin{bmatrix}c_i^{j"} & r_i^{j"}\end{bmatrix}^\top$ which represents the virtual image point after weighting.

$$\begin{bmatrix}c_i^j\\ r_i^j\end{bmatrix} - \begin{bmatrix}c_i^{j"}\\ r_i^{j"}\end{bmatrix} = \mathbf{C}_i^{j\,-1}\left(\begin{bmatrix}c_i^j\\ r_i^j\end{bmatrix} - \begin{bmatrix}c_i^{j'}\\ r_i^{j'}\end{bmatrix}\right), \tag{58}$$

where $\begin{bmatrix}c_i^{j'} & r_i^{j'}\end{bmatrix}^\top$ is the projection image point with image measurement $\begin{bmatrix}c_i^j & r_i^j\end{bmatrix}^\top$, which is defined as:

$$\begin{bmatrix}c_i^{j'} & r_i^{j'}\end{bmatrix}^\top = \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j)). \tag{59}$$

Our goal is to prove that using rectified coordinates as the observed image point will project on the same image point with the normalized measurement-based projection. We can summarize such equivalent as the following equation:

$$\begin{bmatrix}c_i^{j"} & r_i^{j"}\end{bmatrix}^\top = \Pi(\mathbf{R}^j(r_i^{j"})\mathbf{P}_i + \mathbf{t}^j(r_i^{j"})). \tag{60}$$

We follow the schedule that firstly solves the Eq. (58) to get the rectified image coordinate vector $\begin{bmatrix}c_i^{j"} & r_i^{j"}\end{bmatrix}^\top$, use the rectified image coordinate vector in normalized measurement based projection to get the projection point and finally check out whether it is the same.

D.2. New Rectified Image Coordinate Solution

We solve Eq. (58) consequently. Firstly, we solve $r_i^{j"}$.

$$(1 - \boldsymbol{\chi}_i^j(2))(r_i^j - r_i^{j"}) = (r_i^j - r_i^{j'}), \tag{61}$$

$$\begin{aligned}r_i^{j"} &= r_i^j + \frac{r_i^{j'} - r_i^j}{1 - \boldsymbol{\chi}_i^j(2)} \\ &= r_i^{j'} + \frac{\boldsymbol{\chi}_i^j(2)(r_i^{j'} - r_i^j)}{1 - \boldsymbol{\chi}_i^j(2)},\end{aligned} \tag{62}$$

We then substitute $r_i^{j"}$ to solve $c_i^{j"}$.

$$(c_i^j - c_i^{j"}) = \frac{\boldsymbol{\chi}_i^j(1)}{1 - \boldsymbol{\chi}_i^j(2)}(r_i^j - r_i^{j'}) + (c_i^j - c_i^{j'}), \tag{63}$$

$$c_i^{j"} = c_i^{j'} + \frac{\boldsymbol{\chi}_i^j(1)(r_i^{j'} - r_i^j)}{1 - \boldsymbol{\chi}_i^j(2)}. \tag{64}$$

D.3. Proof of equivalent after projection

We then substitute $r_i^{j''}$ and $c_i^{j''}$ in normalized measurement based projection.

$$\begin{aligned}
\begin{bmatrix} c_i^{j\,new} \\ r_i^{j\,new} \end{bmatrix} &= \Pi(\mathbf{R}^j(r_i^{j''})\mathbf{P}_i + \mathbf{t}^j(r_i^{j''})) \\
&= \Pi(\mathbf{R}^j(r_i^j + \frac{r_i^{j'} - r_i^j}{1-\boldsymbol{\chi}_i^j(2)})\mathbf{P}_i + \mathbf{t}^j(r_i^j + \frac{r_i^{j'} - r_i^j}{1-\boldsymbol{\chi}_i^j(2)})) \\
&\approx \begin{bmatrix} c_i^{j'} \\ r_i^{j'} \end{bmatrix} + (\frac{\partial \Pi(\mathbf{R}^j(r_i^j)\mathbf{P}_i + \mathbf{t}^j(r_i^j))}{\partial r_i^j})\frac{r_i^{j'} - r_i^j}{1-\boldsymbol{\chi}_i^j(2)} \\
&= \begin{bmatrix} c_i^{j'} \\ r_i^{j'} \end{bmatrix} + \boldsymbol{\chi}_i^j \frac{r_i^{j'} - r_i^j}{1-\boldsymbol{\chi}_i^j(2)} \\
&= \begin{bmatrix} c_i^{j''} \\ r_i^{j''} \end{bmatrix}
\end{aligned} \tag{65}$$

D.4. Connection between Normalized DC- RSBA and NW-RSBA

From Eq. (65), we can get a such summary that Normalized *DC-RSBA* is equivalent to the proposed *NW-RSBA* mathematical. It is amazing to view that although the two formulations are totally different from each other, they both bring in the implicit rolling shutter constraint to optimization. However, although these two methods are equivalent to each other, our proposed *NW-RSBA* is much easier and faster to solve since we provide detailed analytical Jacobian matrices.

E. Proposed NW-RSBA Algorithm Pipeline

In this section, we provide a detailed bundle adjustment algorithm pipeline with the standard Gauss-Newton least square solver.

List of Algorithms

F. Experimental Settings and Evaluation Metrics

In this section, we provide detailed experiment settings and evaluation metrics used in synthetic data experiments and real data experiments.

F.1. Synthetic Data

Experimental Settings. We simulate 5 RS cameras located randomly on a sphere with a radius of 20 units pointing to a cubical scene with 56 points. The RS image size is 1280 × 1080 px, the focal length is about 1000, and the optical center is at the center of the image domain. We compare all methods by varying the speed, the noise on image measurements, and the readout direction. The results are obtained after collecting the errors over 300 trials each epoch. The default setting is 10 deg/frame and 1 unit/frame for angular and linear velocity, standard covariance noise.

- **Varying Speed**: We evaluate the accuracy of five approaches against increasing angular and linear velocity from 0 to 20 deg/frame and 0 to 2 units/frame gradually, with random directions.

Algorithm 1: Normalized Weighted RSBA

Input: Initial rolling shutter camera poses $\{\mathbf{R}^1, \mathbf{t}^1, \boldsymbol{\omega}^1, \mathbf{d}^1\}$,...,$\{\mathbf{R}^j, \mathbf{t}^j, \boldsymbol{\omega}^j, \mathbf{d}^j\}$, points $\mathbf{P}_1$,...,$\mathbf{P}_i$ as $\boldsymbol{\theta}$ and point measurement in normalized image coordinate $\mathbf{q}_{1\ldots i}^{1\ldots j}$

Output: Refined parameters $\boldsymbol{\theta}^*$

while *(not reach max iteration) and (not satisfy stopping criteria)* **do**
  for *Each camera* $j \in \mathcal{F}$ **do**
    for *Each point* $i \in \mathcal{P}_j$ **do**
      Calculate weighted reprojection error $\hat{\mathbf{e}}_i^j$ using Alg. 2;
      Construct Jacobian matrix $\mathbf{J}_i^j$ using Alg. 3;
      Parallel connect $\mathbf{J}_i^j$ to $\mathbf{J}$;
      Stack $\hat{\mathbf{e}}_i^j$ into $\hat{\mathbf{e}}$;
    end
  end
  Solve normal equation $\mathbf{J}^\top\mathbf{J}\boldsymbol{\delta} = -\mathbf{J}^\top\hat{\mathbf{e}}$ using Alg. 4;
  Update camera poses and points parameters $\boldsymbol{\theta}$ using $\boldsymbol{\delta}$;
end

Algorithm 2: Computation of weighted reprojection error

Input : Rolling shutter camera poses $\{\mathbf{R}^j, \mathbf{t}^j, \boldsymbol{\omega}^j, \mathbf{d}^j\}$, points $\mathbf{P}_i$ and point measurement in normalized image coordinate $\mathbf{q}_i^j$

Output: Normalized weighted error $\hat{\mathbf{e}}_i^j$

Compute weight matrix $\mathbf{C}_i^j$ using Eq. (10);
Compute standard reprojection error $\mathbf{e}_i^j$;
Return normalized weighted reprojection error $\hat{\mathbf{e}}_i^j$ using Eq. (15);

Algorithm 3: Computation of Jacobian matrix

Input : RS camera poses $\{\mathbf{R}^j, \mathbf{t}^j, \boldsymbol{\omega}^j, \mathbf{d}^j\}$, points coordinate $\mathbf{P}_i$ and point measurement in normalized image coordinate $\mathbf{q}_i^j$

Output: Jacobian matrix $\mathbf{J}_i^j$

Calculate $\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{P}_i}$ using Eq. (16);

Calculate $\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \boldsymbol{\xi}^j}$ using Eq. (17);

Calculate $\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{t}^j}$ using Eq. (18);

Calculate $\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \boldsymbol{\omega}^j}$ using Eq. (19);

Calculate $\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{d}^j}$ using Eq. (20);

Construct

$$\begin{aligned}\mathbf{J}_i^j &= \begin{bmatrix}\mathbf{J}_{i,rs}^j & \mathbf{J}_{i,gs}^j & \mathbf{J}_{i,p}^j\end{bmatrix} \\ &= \begin{bmatrix}[\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \boldsymbol{\omega}^j}, \frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{d}^j}] & [\frac{\partial \hat{\mathbf{e}}_i^j}{\partial \boldsymbol{\xi}^j}, \frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{t}^j}] & \frac{\partial \hat{\mathbf{e}}_i^j}{\partial \mathbf{P}_i}\end{bmatrix};\end{aligned}$$

Algorithm 4: Solve the normal equation using two-stage Schur complement

Input : Jacobian matrix $\mathbf{J}$ and weighted error vector $\hat{\mathbf{e}}$

Output: Updated vector $\boldsymbol{\delta}$

Compute Schur complement matrix $\mathbf{S}_p$ and $\mathbf{S}_{rs}$;

Compute auxiliary vectors $\mathbf{t}^*$ and $\mathbf{u}^*$;

Solve normal equation cascadingly:

- Get $\boldsymbol{\delta}_{rs}$ by solving $\mathbf{S}_{rs}\boldsymbol{\delta}_{rs} = -\mathbf{t}^*$;
- Get $\boldsymbol{\delta}_{gs}$ by solving $\mathbf{U}^*\boldsymbol{\delta}_{gs} = -\mathbf{u}^* - \mathbf{S}^{*\top}\boldsymbol{\delta}_{rs}$;
- Get $\boldsymbol{\delta}_p$ by solving $\mathbf{V}\boldsymbol{\delta}_p = -\mathbf{v} - \mathbf{T}^\top\boldsymbol{\delta}_{rs} - \mathbf{W}^\top\boldsymbol{\delta}_{gs}$;
- Stack $\boldsymbol{\delta}_{gs}$ $\boldsymbol{\delta}_{rs}$ $\boldsymbol{\delta}_p$ into $\boldsymbol{\delta}$;

- **Varying Noise Level**: We evaluate the accuracy of five approaches against increasing noise level from 0 to 2 pixels.
- **Varying Readout Direction**: We evaluate the robustness of five methods with an RS critical configuration. Namely, the readout directions of all views are almost parallel. Thus, we vary the readout directions of the cameras from parallel to perpendicular by increasing the angle from 0 to 90 degrees.
- **Runtime**: We compare the time cost of all methods against increasing the number of cameras from 50 - 250 with a fixed number of points.

Evaluation metrics. In this section, we use three metrics to evaluate the performances, namely reconstruction error, rotation error, and translation error.

- **Reconstruction Error $\mathbf{e}_{\text{point}}$**: We use the reconstruction error to measure the difference between computed and ground truth 3D points, which is defined as: $\mathbf{e}_{\text{point}} = \left\|\mathbf{P} - \mathbf{P}_{\text{GT}}\right\|^2$.
- **Rotation Error $\mathbf{e}_{\text{rot}}$**: We utilize the geodesic distance to measure the error between optimized rotation and ground truth. The error is defined as: $\mathbf{e}_{\text{rot}} = \arccos((\text{tr}(\mathbf{R}\mathbf{R}_{\text{GT}}^\top) - 1)/2)$.
- **Translation Error $\mathbf{e}_{\text{trans}}$**: We use normalized inner product distance to measure the error between optimized translation and ground truth, which is defined as: $\mathbf{e}_{\text{trans}} = \arccos(\mathbf{t}^\top\mathbf{t}_{\text{GT}}/(\|\mathbf{t}\|\,\|\mathbf{t}_{\text{GT}}\|))$.

F.2. Real Data

Datasets Settings. We compare all the RSC methods in the following publicly available RS datasets.

- ***WHU-RSVI***: WHU dataset [1] was published in [2] and provided ground truth synthetic GS images, RS images and camera poses.
- ***TUM-RSVI***: The TUM RS dataset[2] was published in [6] and contained time-synchronized global-shutter, and rolling-shutter images captured by a non-perspective camera rig and ground-truth poses recorded by motion capture system for ten RS video sequences.

Evaluation metrics. In this section, we use three metrics to evaluate the performances, namely absolute trajectory error, tracking duration and real-time factor.

- **Absolute trajectory error (ATE).** We use the absolute trajectory error (***ATE***) [6] to evaluate the VO results quantitatively. Given ground truth frame positions $\bar{\mathbf{c}}_i \in \mathbb{R}^3$ and corresponding Orb-SLAM [5] tracking results $\mathbf{c}_i \in \mathbb{R}^3$ using corrected sequence by each RSC method. It is defined as

$$e_{\text{ate}} = \min_{\mathbf{T}\in\text{Sim}(3)} \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{T}(\mathbf{c}_i) - \bar{\mathbf{c}}_i\|}, \tag{66}$$

where $\mathbf{T} \in \text{Sim}(3)$ is a similarity transformation that aligns the estimated trajectory with the ground truth one since the scale is not observable for monocular methods. We run each method 20 times on each sequence to obtain the ATE e_{ate}.

[1] http://aric.whu.edu.cn/caolike/2019/11/05/the-whu-rsvi-dataset/

[2] https://vision.in.tum.de/data/datasets/rolling-shutter-dataset

- **Tracking duration (DUR).** Besides, we find out that some RSC solutions provide the results of corrections that are even worse than the original input RS frames. This leads to failure in tracking and makes Orb-SLAM interrupt before the latest capture frame. Therefore, we use the ratio of the successfully tracked frames out of the total frames ***DUR*** as an evaluation metric.

- **Realtime factor ϵ.** The realtime factor ϵ is calculated as the sequence's actual duration divided by the algorithm's processing time.

References

[1] Cenek Albl, Akihiro Sugimoto, and Tomas Pajdla. Degeneracies in rolling shutter sfm. In *ECCV*, 2016. 3, 8, 9

[2] Like Cao, Jie Ling, and Xiaohui Xiao. The whu rolling shutter visual-inertial dataset. *IEEE Access*, 8:50771–50779, 2020. 6, 9

[3] Yizhen Lao, Omar Ait-Aider, and Helder Araujo. Robustified structure from motion with rolling-shutter camera using straightness constraint. *Pattern Recognition Letters*, 2018. 8

[4] Manolis IA Lourakis and Antonis A Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):1–30, 2009. 8, 9

[5] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *T-RO*, 2015. 6

[6] David Schubert, Nikolaus Demmel, Lukas von Stumberg, Vladyslav Usenko, and Daniel Cremers. Rolling-shutter modelling for direct visual-inertial odometry. In *IROS*, 2019. 6, 9

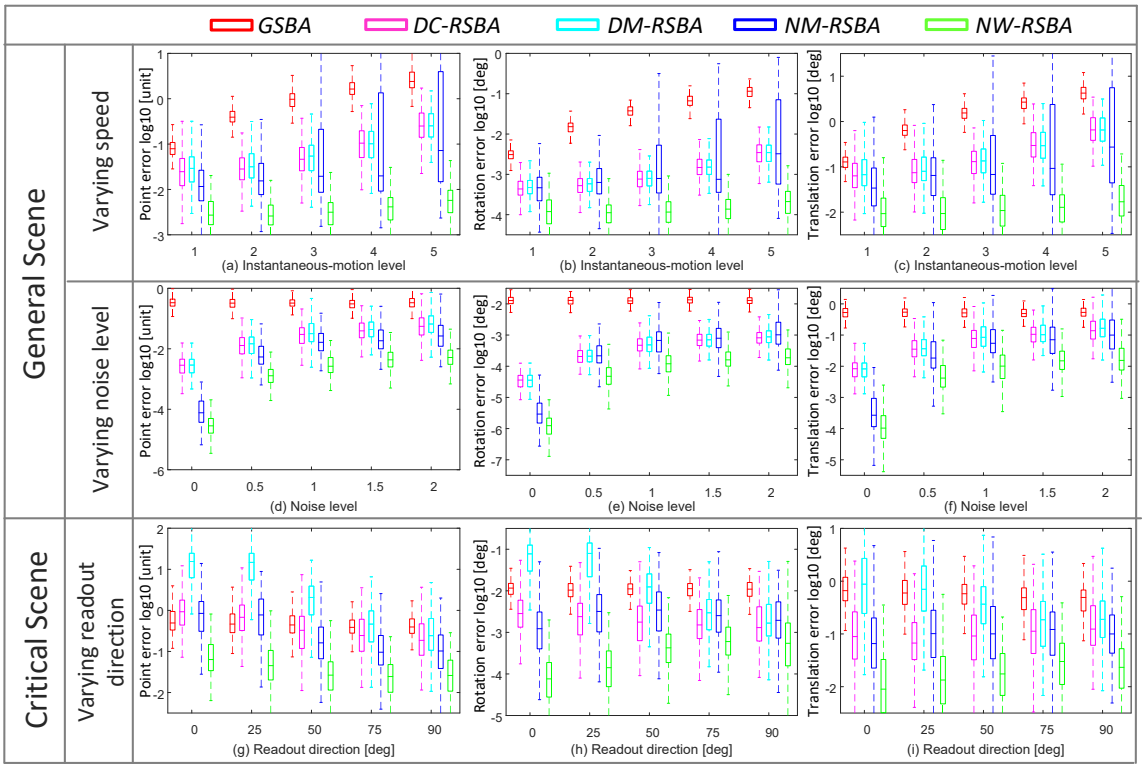

Figure 1. Camera pose (2nd and 3rd columns) and reconstruction (1st column) errors of *GSBA*, *DC-RSBA*, *DM-RSBA*, *NM-RSBA* and *NW-RSBA* with increasing angular and linear velocity (1st row) and noise levels in the image (2nd row) in general scenes, also with increasing readout directions in degeneracy scene (3rd row).

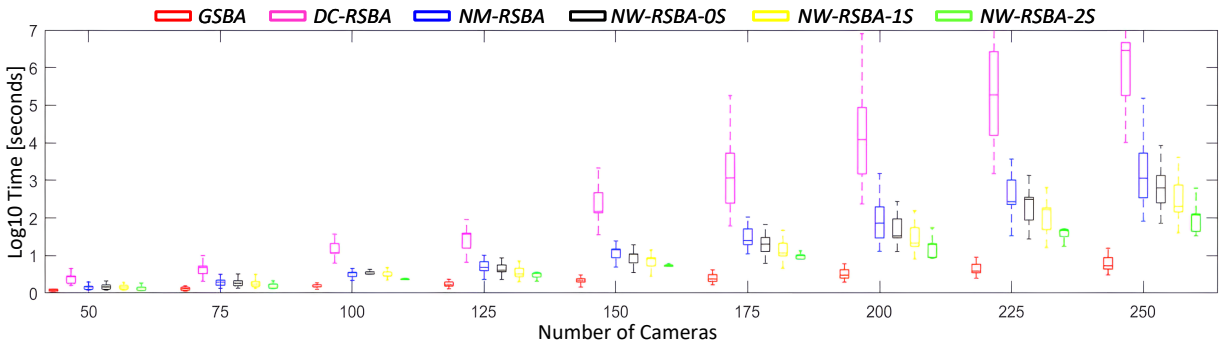

Figure 2. Time cost of *GSBA* [4], *DC-RSBA* [3], *NM-RSBA* [1], *NW-RSBA-0S* (without Schur complement), *NW-RSBA-1S* (one-stage Schur complement to Jacobian matrices with series connection), and proposed *NW-RSBA-2S* (two-stage Schur complement to Jacobian matrices with parallel connection) with increasing camera number.

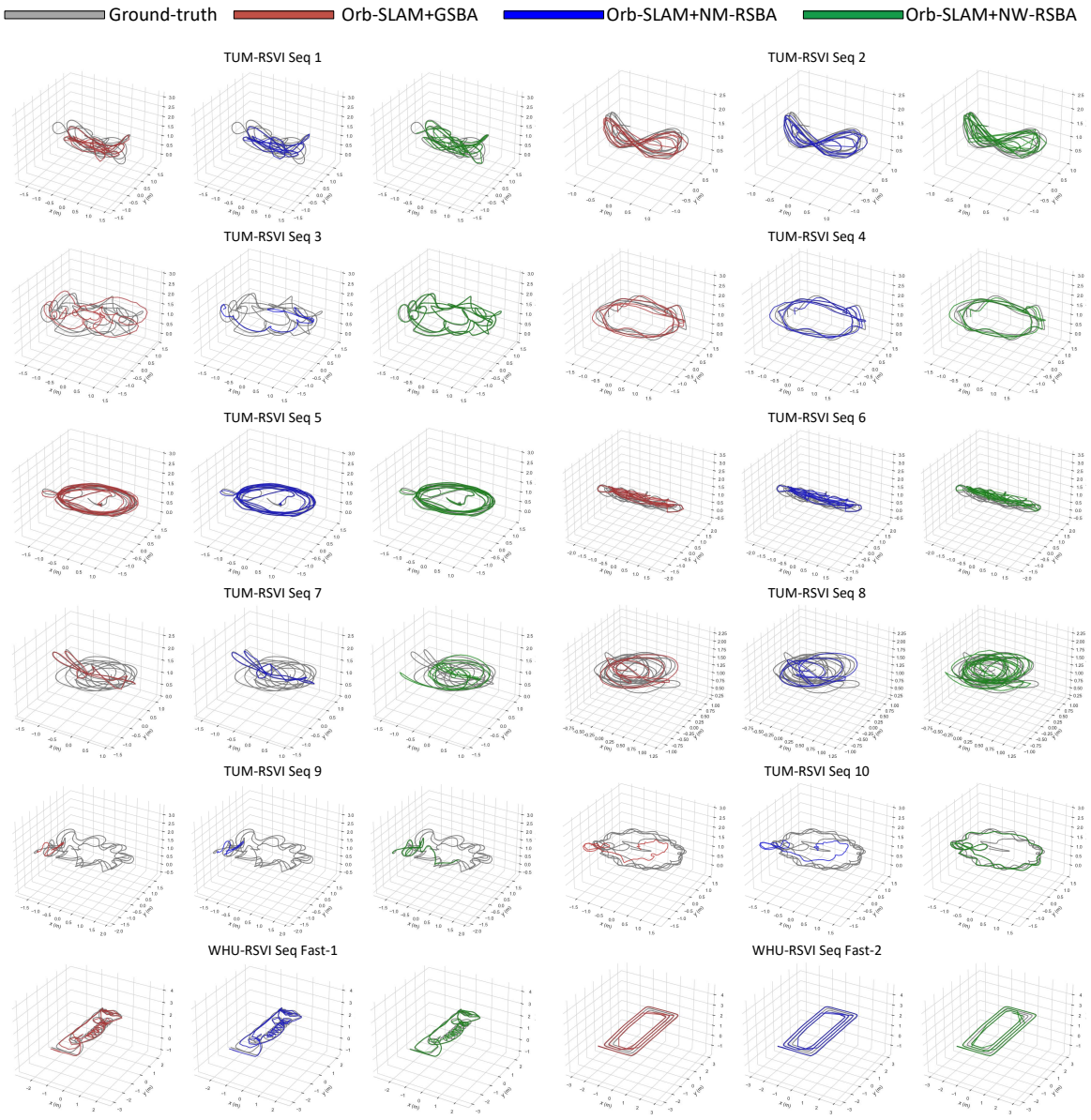

Figure 3. Ground truth and trajectories estimated by *GSBA* [4], *NM-RSBA* [1] and proposed *NW-RSBA* after Sim(3) alignment on 10 sequences from TUM-RSVI [6] and 2 sequences from WHU-RSVI [2] datasets.