Supplementary Material: A Calibration Method for the Generalized Imaging Model with Uncertain Calibration Target Coordinates

David Uhlig and Michael Heizmann

Institute of Industrial Information Technology, Karlsruhe Institute of Technology, Germany
{david.uhlig,michael.heizmann}@kit.edu

1 Supplementary Material

In this supplementary material we provide some details to the calculations from the main paper and we present proofs to some statements. These are intended to help the interested reader to better understand the proposed method.

2 Basics

With $\mathbf{A} \in \mathbb{R}^{k\times l}$, $\mathbf{B} \in \mathbb{R}^{l\times m}$, $\mathbf{C} \in \mathbb{R}^{m\times n}$, using the *Kronecker* product $\otimes$ and the *vec*-operator we can write:

$$\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{B})\,, \tag{1}$$

where $(\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A}) \in \mathbb{R}^{nk\times ml}$ and $\mathrm{vec}(\mathbf{B}) \in \mathbb{R}^{lm}$ and with the *vec*-operator that stacks the columns of matrix $\mathbf{B}$.

The vector cross product can be formulated using the cross-product-operator $[\cdot]_\times$. With $\boldsymbol{\eta}, \boldsymbol{\mu} \in \mathbb{R}^3$:

$$[\boldsymbol{\eta}]_\times = \begin{bmatrix} 0 & -\eta_3 & \eta_2 \\ \eta_3 & 0 & -\eta_1 \\ -\eta_2 & \eta_1 & 0 \end{bmatrix}, \tag{2}$$

$$\boldsymbol{\eta} \times \boldsymbol{\mu} = [\boldsymbol{\eta}]_\times \boldsymbol{\mu} = [\boldsymbol{\mu}]_\times^{\mathrm{T}} \boldsymbol{\eta} = -\boldsymbol{\mu} \times \boldsymbol{\eta} \tag{3}$$

$$\mathrm{vec}([\boldsymbol{\eta}]_\times) = \mathbf{H}\boldsymbol{\eta}\,, \tag{4}$$

$$\mathbf{H} = \left[\mathrm{vec}([\mathbf{e}_1]_\times), \mathrm{vec}([\mathbf{e}_2]_\times), \mathrm{vec}([\mathbf{e}_3]_\times)\right]\,, \tag{5}$$

with the unit basis vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$.

3 Objective function matrices

3.1 Matrices of pose subproblem

In Section 3.2. of the main paper, for every single pose with index k, we obtain an optimization problem with objective function

$$f(\mathbf{R}_k, \mathbf{t}_k) = \sum_i w_{ik} \left\|(\mathbf{R}_k\mathbf{x}_{ik} + \mathbf{t}_k) \times \mathbf{d}_i - \mathbf{m}_i\right\|^2\,. \tag{6}$$

We can write this objective function in a more compact form by using the *Kronecker* identity (1), the cross product operator (2), the *vec*-operator with $\mathbf{r}_k = \mathrm{vec}(\mathbf{R}_k)$, and the introduction of some new variables:

$$\mathbf{A}_{\mathrm{rr},k} = \sum_i w_{ik} \left(\mathbf{x}_{ik}\mathbf{x}_{ik}^{\mathrm{T}}\right) \otimes \left([\mathbf{d}_i]_\times [\mathbf{d}_i]_\times^{\mathrm{T}}\right) , \tag{7}$$

$$\mathbf{A}_{\mathrm{tt},k} = \sum_i w_{ik} [\mathbf{d}_i]_\times [\mathbf{d}_i]_\times^{\mathrm{T}} , \tag{8}$$

$$\mathbf{A}_{\mathrm{tr},k} = \sum_i 2w_{ik} [\mathbf{d}_i]_\times \left(\mathbf{x}_{ik}^{\mathrm{T}} \otimes [\mathbf{d}_i]_\times^{\mathrm{T}}\right) , \tag{9}$$

$$\mathbf{b}_{\mathrm{r},k} = \sum_i -2w_{ik} \left(\mathbf{x}_{ik}^{\mathrm{T}} \otimes [\mathbf{d}_i]_\times^{\mathrm{T}}\right)^{\mathrm{T}} \mathbf{m}_i , \tag{10}$$

$$\mathbf{b}_{\mathrm{t},k} = \sum_i 2w_{ik} [\mathbf{d}_i]_\times^{\mathrm{T}} \mathbf{m}_i , \tag{11}$$

$$h_k = \sum_i w_{ik} \|\mathbf{m}_i\|^2 , \tag{12}$$

which results in the more compact form:

$$f(\mathbf{r}_k, \mathbf{t}_k) = \mathbf{r}_k^{\mathrm{T}} \mathbf{A}_{\mathrm{rr},k} \mathbf{r}_k + \mathbf{t}_k^{\mathrm{T}} \mathbf{A}_{\mathrm{tt},k} \mathbf{t}_k + \mathbf{t}_k^{\mathrm{T}} \mathbf{A}_{\mathrm{tr},k} \mathbf{r}_k + \mathbf{b}_{\mathrm{r},k}^{\mathrm{T}} \mathbf{r}_k + \mathbf{b}_{\mathrm{t},k}^{\mathrm{T}} \mathbf{t}_k + h_k . \tag{13}$$

3.2 Matrices of rotation subproblem

In Section 3.2. of the main paper, we obtain an optimization problem for the rotation estimation:

$$f(\mathbf{R}) = \mathbf{r}^{\mathrm{T}} \mathbf{A} \mathbf{r} + \mathbf{b}^{\mathrm{T}} \mathbf{r} + c , \quad \text{s.t. } \mathbf{r} = \mathrm{vec}(\mathbf{R}) , \mathbf{R} \in \mathrm{SO}(3) . \tag{14}$$

with:

$$\mathbf{A} = \mathbf{A}_{\mathrm{rr}} - \frac{1}{4} \mathbf{A}_{\mathrm{tr}}^{\mathrm{T}} \mathbf{A}_{\mathrm{tt}}^{-1} \mathbf{A}_{\mathrm{tr}} , \tag{15}$$

$$\mathbf{b} = \mathbf{b}_{\mathrm{r}} - \frac{1}{4} \mathbf{A}_{\mathrm{tr}}^{\mathrm{T}} \mathbf{A}_{\mathrm{tt}}^{-1} \mathbf{b}_{\mathrm{t}} , \tag{16}$$

$$c = h - \frac{1}{4} \mathbf{b}_{\mathrm{t}}^{\mathrm{T}} \mathbf{A}_{\mathrm{tt}}^{-1} \mathbf{b}_{\mathrm{t}} . \tag{17}$$

4 Rotation optimization

In Section 3.2 we needed to minimize:

$$f(\mathbf{R}) = \mathbf{r}^{\mathrm{T}} \mathbf{A} \mathbf{r} + \mathbf{b}^{\mathrm{T}} \mathbf{r} + c , \quad \text{s.t. } \mathbf{r} = \mathrm{vec}(\mathbf{R}) , \mathbf{R} \in \mathrm{SO}(3) , \tag{18}$$

to find an optimal rotation matrix $\mathbf{R}$. We explained that we use a gradient-based optimization on the rotation manifold. The basic theory and the methods to calculate the gradient and the Hessian for such an optimization problem can be found in the literature [1–3]. In the following, we give a short introduction and demonstrate how the gradient and Hessian for our optimization problem (18) can be calculated. Fig. 1 visualizes the procedure.

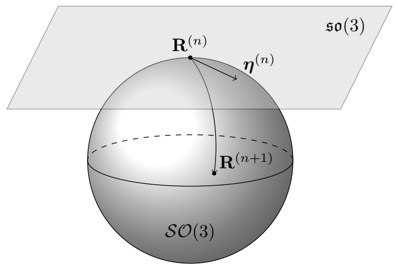

Fig. 1: Visualization of the local parametrization of the $\mathrm{SO}(3)$-manifold through its tangent space. The search direction is found in the $\mathfrak{so}(3)$ tangent space and back-projected onto the manifold to find a minimum of the objective.

4.1 Gradient

In the $\mathfrak{so}(3)$-tangent space we calculate the derivative in direction $\boldsymbol{\eta}$. With $\mathbf{R}(\boldsymbol{\eta}) = \mathrm{e}^{[\boldsymbol{\eta}]_\times}\mathbf{R}$, $\mathbf{r} = \mathrm{vec}(\mathbf{R})$ and $\mathbf{H}$ as defined in (5), we get:

$$Df_{\boldsymbol{\eta}}(\mathbf{R})[\boldsymbol{\eta}] = \frac{\partial}{\partial\varepsilon} f_{\boldsymbol{\eta}\varepsilon}(\mathbf{R})\Big|_{\varepsilon=0}, \tag{19}$$

$$\frac{\partial}{\partial\varepsilon} f_{\boldsymbol{\eta}\varepsilon}(\mathbf{R}) = \frac{\partial\mathbf{r}(\boldsymbol{\eta}\varepsilon)^{\mathrm{T}}}{\partial\varepsilon}\frac{\partial}{\partial\mathbf{r}} f(\mathbf{R})\Big|_{\mathbf{r}=\mathbf{r}(\boldsymbol{\eta}\varepsilon)} = \frac{\partial\mathbf{r}(\boldsymbol{\eta}\varepsilon)^{\mathrm{T}}}{\partial\varepsilon} 2\left(\mathbf{A}\mathbf{r}(\boldsymbol{\eta}\varepsilon) + \mathbf{b}\right) \tag{20}$$

$$= 2\mathrm{vec}([\boldsymbol{\eta}]_\times \mathrm{e}^{\varepsilon[\boldsymbol{\eta}]_\times}\mathbf{R})^{\mathrm{T}}\left(\mathbf{A}\mathbf{r}(\boldsymbol{\eta}\varepsilon) + \mathbf{b}\right). \tag{21}$$

With $\varepsilon \to 0$ it follows:

$$Df_{\boldsymbol{\eta}}(\mathbf{R})[\boldsymbol{\eta}] = \frac{\partial}{\partial\varepsilon} f_{\boldsymbol{\eta}\varepsilon}(\mathbf{R})\Big|_{\varepsilon=0} = 2\mathrm{vec}([\boldsymbol{\eta}]_\times\mathbf{R})^{\mathrm{T}}(\mathbf{A}\mathbf{r} + \mathbf{b}) \tag{22}$$

$$\stackrel{(1)}{=} \left(\left(\mathbf{R}^{\mathrm{T}} \otimes \mathbf{I}\right)\mathrm{vec}([\boldsymbol{\eta}]_\times)\right)^{\mathrm{T}}(\mathbf{A}\mathbf{r} + \mathbf{b}) \tag{23}$$

$$\stackrel{(4)}{=} 2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}(\mathbf{R} \otimes \mathbf{I})(\mathbf{A}\mathbf{r} + \mathbf{b}) = \boldsymbol{\eta}^{\mathrm{T}}\mathrm{grad}(f). \tag{24}$$

Finally, we obtain the gradient of our locally parameterized objective function:

$$\mathrm{grad}(f) = 2\mathbf{H}^{\mathrm{T}}(\mathbf{R} \otimes \mathbf{I})(\mathbf{A}\mathbf{r} + \mathbf{b}). \tag{25}$$

4.2 Hessian

Similar to the previous calculations, we calculate the second order derivative:

$$D\,\mathrm{grad}(f)[\boldsymbol{\eta}] = \boldsymbol{\eta}^{\mathrm{T}}\frac{\partial}{\partial\varepsilon}\mathrm{grad}(f)\Big|_{\varepsilon=0}, \tag{26}$$

$$\boldsymbol{\eta}^{\mathrm{T}}\frac{\partial}{\partial\varepsilon}\mathrm{grad}(f) = \boldsymbol{\eta}^{\mathrm{T}}\frac{\partial}{\partial\varepsilon} 2\mathbf{H}^{\mathrm{T}}(\mathbf{R}(\boldsymbol{\eta}\varepsilon) \otimes \mathbf{I})(\mathbf{A}\mathbf{r}(\boldsymbol{\eta}\varepsilon) + \mathbf{b}) \tag{27}$$

$$= \frac{\partial}{\partial\varepsilon}\left(2\mathrm{vec}([\boldsymbol{\eta}]_\times \mathrm{e}^{\varepsilon[\boldsymbol{\eta}]_\times}\mathbf{R})^{\mathrm{T}}(\mathbf{A}\mathrm{vec}(\mathrm{e}^{\varepsilon[\boldsymbol{\eta}]_\times}\mathbf{R}) + \mathbf{b})\right). \tag{28}$$

With $\varepsilon \to 0$ it follows:

$$D\,\mathrm{grad}(f)[\boldsymbol{\eta}] = 2\mathrm{vec}([\boldsymbol{\eta}]_\times^2\,\mathbf{R})^{\mathrm{T}}\left(\mathbf{Ar}+\mathbf{b}\right) + 2\mathrm{vec}([\boldsymbol{\eta}]_\times\,\mathbf{R})^{\mathrm{T}}\mathbf{A}\mathrm{vec}([\boldsymbol{\eta}]_\times\,\mathbf{R})\,. \quad (29)$$

It follows with the reshape operator $\mathrm{mat}(\mathrm{vec}(\mathbf{A})) = \mathbf{A}$:

$$\begin{aligned}
2\mathrm{vec}([\boldsymbol{\eta}]_\times^2\,\mathbf{R})^{\mathrm{T}}\left(\mathbf{Ar}+\mathbf{b}\right) &= 2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left([\boldsymbol{\eta}]_\times\,\mathbf{R}\otimes\mathbf{I}\right)\left(\mathbf{Ar}+\mathbf{b}\right)\\
&= 2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathrm{vec}\left(\mathrm{mat}\left(\mathbf{Ar}+\mathbf{b}\right)\mathbf{R}^{\mathrm{T}}\,[\boldsymbol{\eta}]_\times^{\mathrm{T}}\right)\\
&= 2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{I}\otimes\mathrm{mat}\left(\mathbf{Ar}+\mathbf{b}\right)\mathbf{R}^{\mathrm{T}}\right)\mathrm{vec}([\boldsymbol{\eta}]_\times^{\mathrm{T}})\\
&= -2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{I}\otimes\mathrm{mat}\left(\mathbf{Ar}+\mathbf{b}\right)\mathbf{R}^{\mathrm{T}}\right)\mathbf{H}\boldsymbol{\eta}\\
&= \boldsymbol{\eta}^{\mathrm{T}}\mathrm{Hess}_1(f)\boldsymbol{\eta}\,,\\
2\mathrm{vec}([\boldsymbol{\eta}]_\times\,\mathbf{R})^{\mathrm{T}}\mathbf{A}\mathrm{vec}([\boldsymbol{\eta}]_\times\,\mathbf{R}) &= 2\boldsymbol{\eta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{R}\otimes\mathbf{I}\right)\mathbf{A}\left(\mathbf{R}\otimes\mathbf{I}\right)^{\mathrm{T}}\mathbf{H}\boldsymbol{\eta}\\
&= \boldsymbol{\eta}^{\mathrm{T}}\mathrm{Hess}_2(f)\boldsymbol{\eta}\,.
\end{aligned}$$

Finally, we obtain the Hessian of our locally parameterized objective function:

$$\mathrm{Hess}(f) = \mathrm{Hess}_1(f) + \mathrm{Hess}_2(f) \quad (30)$$

$$= -2\mathbf{H}^{\mathrm{T}}\left(\mathbf{I}\otimes\mathrm{mat}\left(\mathbf{Ar}+\mathbf{b}\right)\mathbf{R}^{\mathrm{T}}\right)\mathbf{H} + 2\mathbf{H}^{\mathrm{T}}\left(\mathbf{R}\otimes\mathbf{I}\right)\mathbf{A}\left(\mathbf{R}\otimes\mathbf{I}\right)^{\mathrm{T}}\mathbf{H}\,. \quad (31)$$

5 Proof: Invertibility of $\mathbf{A}_{\mathrm{tt}}$

Calculating the translation vector from the rotation in Section 3.2. required the matrix $\mathbf{A}_{\mathrm{tt}}$ to be invertible. Here, we show that $\mathbf{A}_{\mathrm{tt}}$ is positive definite in most cases and thus invertible. We need to show:

$$\mathbf{x}^T\mathbf{A}_{\mathrm{tt}}\mathbf{x} > 0 \implies \mathbf{A}_{\mathrm{tt}} \text{ is invertible.} \quad (32)$$

With $\|\mathbf{d}_i\| = 1, w_{ik} > 0$ and $\forall\mathbf{x}\in\mathbb{R}^3$ with $\|\mathbf{x}\| > 0$ we calculate:

$$\begin{aligned}
\mathbf{x}^T\mathbf{A}_{\mathrm{tt}}\mathbf{x} = \mathbf{x}^T\sum_i w_{ik}\,[\mathbf{d}_i]_\times\,[\mathbf{d}_i]_\times^T\,\mathbf{x} &= \sum_i w_{ik}\mathbf{x}^T\,[\mathbf{d}_i]_\times\,[\mathbf{d}_i]_\times^T\,\mathbf{x}\\
&= \sum_i w_{ik}\left([\mathbf{d}_i]_\times^T\,\mathbf{x}\right)^T[\mathbf{d}_i]_\times^T\,\mathbf{x} = \sum_i w_{ik}\left\|[\mathbf{d}_i]_\times^T\,\mathbf{x}\right\|^2\\
&= \sum_i w_{ik}\left\|\mathbf{x}\times\mathbf{d}_i\right\|^2 > 0\,.
\end{aligned}$$

This is always true, except for the degenerate case of parallel rays (orthographic projection, *e.g.* telecentric optics). Then $\mathbf{x} = s\mathbf{d}_i\,, \forall i$ with some arbitrary scalar s, results in $\mathbf{x}^T\mathbf{A}_{\mathrm{tt}}\mathbf{x} = 0$. In this case we have an ambiguity in the translation term, because it is not possible to estimate the distance of the calibration pattern to a camera with orthographic projection:

$$\mathbf{t} = \mathbf{t}_0 + s\mathbf{d}_0\,. \quad (33)$$

6 Proof: Convergence of AM-Calibration

Following the research in the field of AM [4, 5], we proof that the proposed alternating minimization technique for camera calibration is convergent, thus

$$f\left(\mathcal{P}^{(n+1)}, \mathcal{L}^{(n+1)}\right) < f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right) , \tag{34}$$

with the current estimate of the ray parameters $\mathcal{L}^{(n)}$, the current estimate of the poses $\mathcal{P}^{(n)} = \left[\mathcal{R}^{(n)}, \mathcal{T}^{(n)}\right]$ and the abbreviation $\mathcal{R} := \{\mathbf{R}_1, \mathbf{R}_2, \dots\}$, $\mathcal{T} := \{\mathbf{t}_1, \mathbf{t}_2, \dots\}$, $\mathcal{L} := \{\mathbf{L}_1, \mathbf{L}_2, \dots\}$.

Define the operators $\mathcal{S}_\mathbf{L}$ and $\mathcal{S}_\mathbf{P}$, as solution to the ray subproblem of Section 3.1. and as solution to the pose subproblem of Section 3.2., respectively:

$$\mathcal{S}_\mathbf{L}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right)\right\} = f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right) , \tag{35}$$

$$\mathcal{S}_\mathbf{P}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right)\right\} = f\left(\mathcal{P}^{(n+1)}, \mathcal{L}^{(n+1)}\right) . \tag{36}$$

Because the optimization of camera rays delivers an optimal solution to its subproblem, we cannot get an increase in the objective function:

$$\mathcal{S}_\mathbf{L}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right)\right\} \leq f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right) . \tag{37}$$

Furthermore, if we initialize the Newton descend algorithm for pose estimation with the previous pose, we always get a descend in the objective function value:

$$\mathcal{S}_\mathbf{P}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right)\right\} < f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right) . \tag{38}$$

In conclusion we obtain:

$$\begin{aligned} f\left(\mathcal{P}^{(n+1)}, \mathcal{L}^{(n+1)}\right) &= \mathcal{S}_\mathbf{P}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right)\right\} \\ &< f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n+1)}\right) \\ &= \mathcal{S}_\mathbf{L}\left\{f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right)\right\} \\ &\leq f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right) , \end{aligned} \tag{39}$$

$$\implies f\left(\mathcal{P}^{(n+1)}, \mathcal{L}^{(n+1)}\right) < f\left(\mathcal{P}^{(n)}, \mathcal{L}^{(n)}\right) \quad \text{q.e.d.} \tag{40}$$

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)
2. Sarkis, M., Diepold, K.: Camera-pose estimation via projective newton optimization on the manifold. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society **21** (2012) 1729–1741

3. Boumal, N.: Optimization and estimation on manifolds. PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium (2014)
4. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear gauss–seidel method under convex constraints. Operations Research Letters **26** (2000) 127–136
5. Niesen, U., Shah, D., Wornell, G.: Adaptive alternating minimization algorithms. In: IEEE International Symposium on Information Theory, 2007, Piscataway, NJ, IEEE Service Center (2007) 1641–1645