Optimal Calibration of a Camera System

Karlheinz Spindler, Fachhochschule Wiesbaden
Kurt-Schumacher-Ring 18, D-65197 Wiesbaden, Germany

Abstract

The calibration of a camera system is modelled as a parameter estimation problem on a nonlinear state space. Properly taking into account the manifold structure of the state space, a calibration algorithm with extremely good convergence properties is derived.

Key words: Camera calibration, parameter estimation, system identification, nonlinear state space, statistics on manifolds.

1. Introduction

We discuss the problem of calibrating a system made up of an arbitrary number of cameras each of which is, possibly after preprocessing, described by a pinhole model. If a number of landmarks can be identified on images taken by these cameras, the resulting image coordinates can be measured. The task is to deduce from these measurements the locations and spatial orientations of the cameras and to assess the accuracies of these data in terms of the (assumed) measurement accuracies and also of the accuracies of the landmark coordinates if these cannot be assumed as perfectly known.

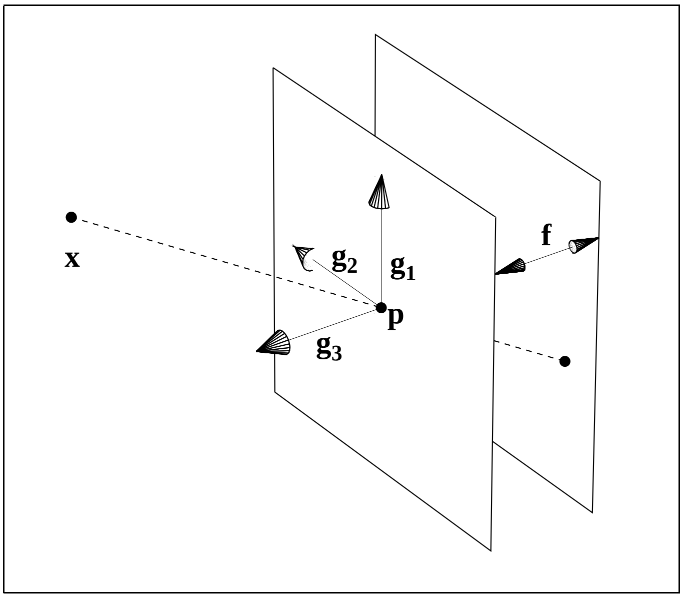

Figure 1: Camera model.

2. Camera System Model

We consider a pinhole camera with focal width f whose optical centre is located at a point p and whose spatial orientation is given by a right-handed orthonormal system (g_1, g_2, g_3) of vectors such that g_1 and g_2 span the image plane whereas g_3 is perpendicular to the image plane, pointing towards the objects of which pictures are taken. We call the matrix $g := (g_1 \mid g_2 \mid g_3)$ whose columns are the vectors g_i the *camera attitude*; this is a rotation matrix, i.e., an element of the rotation group $G = \mathrm{SO}(3)$. Clearly, $g_i = ge_i$ for $i = 1, 2, 3$, where (e_1, e_2, e_3) is the reference coordinate system used. If a picture of a landmark located at a point x is taken, then the light ray from x through p intersects the image plane; i.e., there are real numbers $\lambda > 0$, u and v such that the equation $x + \lambda(p-x) = p + ug_1 + vg_2 - fg_3$ holds, i.e., we have

$$(1) \qquad (\lambda - 1)(p - x) = ug_1 + vg_2 - fg_3$$

(where u and v are the horizontal and vertical image coordinates). Taking the inner product of (1) with g_3 we find that $(\lambda-1)\langle p-x, g_3\rangle = -f$ and hence that

$$(2) \qquad \lambda - 1 = \frac{-f}{\langle p-x, g_3\rangle}.$$

Taking the inner product of (1) with both g_1 and g_2, plugging in (2) and writing $g_i = ge_i$ for $1 \le i \le 3$, we obtain the equations

$$(3) \qquad \begin{aligned} u &= -f \cdot \frac{\langle p-x, ge_1\rangle}{\langle p-x, ge_3\rangle} =: U(p,g;x,f), \\ v &= -f \cdot \frac{\langle p-x, ge_2\rangle}{\langle p-x, ge_3\rangle} =: V(p,g;x,f). \end{aligned}$$

3. Sensitivity of Measurements

To see how sensitively the measurement functions U and V depend on their arguments, we need to calculated the associated partial derivatives. A straightforward calculation shows that the gradients of U and V with respect to p are given by

$$(4) \qquad \begin{aligned} \nabla_p U &= -f \cdot \frac{\langle p-x, g_3\rangle g_1 - \langle p-x, g_1\rangle g_3}{\langle p-x, g_3\rangle^2} \\ &= -f \cdot \frac{(p-x) \times (g_1 \times g_3)}{\langle p-x, g_3\rangle^2} \\ &= f \cdot \frac{(p-x) \times g_2}{\langle p-x, g_3\rangle^2} \end{aligned}$$

and

$$(5) \qquad \begin{aligned} \nabla_p V &= -f \cdot \frac{\langle p-x, g_3\rangle g_2 - \langle p-x, g_2\rangle g_3}{\langle p-x, g_3\rangle^2} \\ &= -f \cdot \frac{(p-x) \times (g_2 \times g_3)}{\langle p-x, g_3\rangle^2} \\ &= -f \cdot \frac{(p-x) \times g_1}{\langle p-x, g_3\rangle^2} \end{aligned}$$

whereas obviously

$$\nabla_x U = -\nabla_p U \quad \text{and} \quad \nabla_x V = -\nabla_p V. \tag{6}$$

To calculate the partial derivatives of U and V with respect to g, we temporarily ignore the fact that g must be an element of $G = \mathrm{SO}(3)$, but treat g simply as an element of the linear space $\mathbb{R}^{3\times 3}$ of all real (3×3)-matrices. Endowing this space with the Frobenius inner product $\langle\!\langle A, B\rangle\!\rangle := \mathrm{tr}(A^T B)$ and using the fact that $\langle u, Av\rangle = \langle\!\langle u\otimes v, A\rangle\!\rangle$ for all $u, v\in\mathbb{R}^3$ and all $A\in\mathbb{R}^{3\times 3}$, we find that

$$\nabla_g U = -f\cdot\frac{\langle p-x, g_3\rangle(p-x)\otimes e_1 - \langle p-x, g_1\rangle(p-x)\otimes e_3}{\langle p-x, g_3\rangle^2}$$
$$\nabla_g V = -f\cdot\frac{\langle p-x, g_3\rangle(p-x)\otimes e_2 - \langle p-x, g_2\rangle(p-x)\otimes e_3}{\langle p-x, g_3\rangle^2}$$

where the gradients are formed with respect to the inner product $\langle\!\langle\cdot,\cdot\rangle\!\rangle$. Letting

$$y \;:=\; g^{-1}(p-x)\,, \tag{7}$$

this can be rewritten as

$$\begin{aligned}\Delta_g U &= \frac{-f}{\langle p-x, g_3\rangle^2}(p-x)\otimes\Big(\langle p-x, g_3\rangle e_1 - \langle p-x, g_1\rangle e_3\Big)\\ &= \frac{-f}{\langle y, e_3\rangle^2}(p-x)\otimes\big(\langle y, e_3\rangle e_1 - \langle y, e_1\rangle e_3\big)\\ &= \frac{-f}{\langle y, e_3\rangle^2}(p-x)\otimes\big(y\times(e_1\times e_3)\big)\end{aligned}$$

and

$$\begin{aligned}\Delta_g V &= \frac{-f}{\langle p-x, g_3\rangle^2}(p-x)\otimes\Big(\langle p-x, g_3\rangle e_2 - \langle p-x, g_2\rangle e_3\Big)\\ &= \frac{-f}{\langle y, e_3\rangle^2}(p-x)\otimes\big(\langle y, e_3\rangle e_2 - \langle y, e_2\rangle e_3\big)\\ &= \frac{-f}{\langle y, e_3\rangle^2}(p-x)\otimes\big(y\times(e_2\times e_3)\big)\end{aligned}$$

so that

$$\begin{aligned}\nabla_g U &= \frac{f}{\langle y, e_3\rangle^2}(p-x)\otimes(y\times e_2),\\ \nabla_g V &= \frac{-f}{\langle y, e_3\rangle^2}(p-x)\otimes(y\times e_1).\end{aligned} \tag{8}$$

Finally,

$$\begin{aligned}\frac{\partial U}{\partial f} &= -\frac{\langle p-x, g_1\rangle}{\langle p-x, g_3\rangle} = \frac{U}{f}\,,\\ \frac{\partial V}{\partial f} &= -\frac{\langle p-x, g_2\rangle}{\langle p-x, g_3\rangle} = \frac{V}{f}\,.\end{aligned} \tag{9}$$

We now use the partial derivatives to show which changes in the measurement functions U and V are caused by changes in the system parameters. The only parameter for which this is not straightforward is the camera attitude g, because now we have to incorporate the constraints $g^T g = \mathbf{1}$ and $\det(g) = 1$. (In other words, the argument g of the functions U and V must not be considered as an element of the linear manifold $\mathbb{R}^{3\times 3}$, but as an element of the nonlinear manifold $G = \mathrm{SO}(3)$.) One possible way of proceeding would be to choose a parametrisation of G, for example Euler angles, which is smoothly invertible about the actual attitude g, and then to calculate the partial derivatives with respect to the chosen coordinates. Proceeding this way, however, may lead to numerical difficulties even if the (unavoidable) singularities of the parametrisation are relatively far away from the actual attitude g; therefore, we choose a different approach. Namely, we consider only increments $\delta g\in\mathbb{R}^{3\times 3}$ which are tangent to G at g (with the consequence that $g+\delta g$ is an element of $G = \mathrm{SO}(3)$ up to second-order effects). This means that we consider only increments of the form

$$\delta g \;=\; gL(\Delta) \quad (\Delta\in\mathbb{R}^3) \tag{10}$$

where, in general, for any given vector $\omega\in\mathbb{R}^3$ the matrix $L(\omega)\in\mathbb{R}^{3\times 3}$ is defined by $L(\omega)a = \omega\times a$; in coordinates, we have

$$L(\omega) \;=\; \begin{bmatrix} 0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}. \tag{11}$$

Now changing g by an increment $\delta g = gL(\Delta)$ yields, in first-order approximation, changes $(\delta U)_g = \langle\!\langle\nabla_g U, \delta g\rangle\!\rangle$ and $(\delta V)_g = \langle\!\langle\nabla_g V, \delta g\rangle\!\rangle$ in U and V, respectively, which – according to (8) – are given by

$$\begin{aligned}(\delta U)_g &= \frac{f}{\langle y, e_3\rangle^2}\langle\!\langle (p-x)\otimes(y\times e_2), \delta g\rangle\!\rangle\\ &= \frac{f}{\langle y, e_3\rangle^2}\langle p-x, (\delta g)(y\times e_2)\rangle\\ &= \frac{f}{\langle y, e_3\rangle^2}\langle p-x, g\big(\Delta\times(y\times e_2)\big)\rangle\\ &= \frac{f}{\langle y, e_3\rangle^2}\langle y, \Delta\times(y\times e_2)\rangle\\ &= \frac{-f}{\langle y, e_3\rangle^2}\langle y\times(y\times e_2), \Delta\rangle\end{aligned} \tag{12}$$

and

$$\begin{aligned}(\delta V)_g &= \frac{-f}{\langle y, e_3\rangle^2}\langle\!\langle (p-x)\otimes(y\times e_1), \delta g\rangle\!\rangle\\ &= \frac{-f}{\langle y, e_3\rangle^2}\langle p-x, (\delta g)(y\times e_1)\rangle\\ &= \frac{-f}{\langle y, e_3\rangle^2}\langle p-x, g\big(\Delta\times(y\times e_1)\big)\rangle\\ &= \frac{-f}{\langle y, e_3\rangle^2}\langle y, \Delta\times(y\times e_1)\rangle\\ &= \frac{f}{\langle y, e_3\rangle^2}\langle y\times(y\times e_1), \Delta\rangle\,.\end{aligned} \tag{13}$$

4. Estimation Procedure

The practical determination of the camera configuration from the available measurements requires filtering out the noise with which the measurements are fraught. This is a standard estimation problem which, in reasonable generality, can be formulated as follows. (See [2], pp. 120-133 for more details.) A measurement vector μ depends on two kinds of parameters U and u which are distinguished because of the different roles they play in the subsequent estimation process: U is treated as a *solve-for parameter* whereas u is taken as a *consider parameter*; i.e., the value of U will be estimated whereas u is only considered in assessing the accuracy of the estimate obtained for U. (In our case $U = (p_1, g_1, \ldots, p_N, g_N)$ while $u = (x_1, \ldots, x_m)$ where x_i is the position of the i-th landmark. The focal widths $f_1, \ldots, f_N$ are assumed to be perfectly known, but could alternatively also be treated as consider parameters.) If $U^\star$ and $u^\star$ are the true (but unknown) parameter values then the measurement vector $\widehat{\mu}$ obtained is

$$(14) \qquad \widehat{\mu} \;=\; \mu(U^\star, u^\star) + n$$

where n is the measurement noise (whose covariance matrix is supposed to be known). We assume that we have initial estimates U_{init} and u_{init} for the parameters in question. While the estimate for u is never changed, we want to iteratively improve the available estimate for U. Thus we ask how to optimally update an "old" estimate U_{old} to obtain a "new" estimate

$$(15) \qquad U_{\text{new}} \;=\; U_{\text{old}} + \delta U\,.$$

To assess the quality of an arbitrary estimate (U, u), we introduce the *residual vector*

$$(16) \qquad \rho(U,u) \;=\; \widehat{\mu} - \mu(U,u)$$

which is a list of the differences between the actually obtained and the theoretically expected measurements. To properly measure the size of the residual vector, we weight the different measurements according to their respective accuracies; i.e., we introduce the scalar quantity

$$(17) \qquad Q(U,u) \;:=\; \rho(U,u)^T W \rho(U,u)$$

with the weighting matrix

$$(18) \qquad W \;=\; \mathrm{Cov}[n]^{-1}\,.$$

Denoting by $\sqrt{W}$ the unique upper triangular matrix M such that $W = M^T M$ (obtained by performing the *Cholesky decomposition* of W; see [1], pp. 37-43, and [4], pp. 146-149) we can write

$$(19) \qquad Q(U,u) \;=\; \|\, \sqrt{W}\rho(U,u) \,\|^2;$$

thus in the case of uncorrelated measurements Q is simply the sum of the squares of the weighted residuals, where the weighting factor for any measurement is the reciprocal of the standard deviation of this measurement. Now an update step δU as in (15) is considered optimal if it minimises the size of the resulting "new" residual vector

$$(20) \qquad \begin{aligned} \rho_{\text{new}} &= \rho(U_{\text{new}}, u_{\text{init}}) \;=\; \rho(U_{\text{old}} + \delta U, u_{\text{init}}) \\ &\approx \rho(U_{\text{old}}, u_{\text{init}}) + (\partial\rho/\partial U)(U_{\text{old}}, u_{\text{init}})\delta U \\ &= \rho_{\text{old}} - A(U_{\text{old}}, u_{\text{init}})\delta U \end{aligned}$$

where

$$(21) \qquad A(U,u) \;:=\; \frac{\partial \mu}{\partial U}(U,u)$$

denotes the matrix of partial derivatives of the measurements with respect to the solve-for parameters. Thus, using first-order approximations, we want to choose the update δU such that

$$(22) \qquad Q_{\text{new}} = \|\, \sqrt{W}\rho_{\text{new}} \,\|^2 = \|\, \sqrt{W}\rho_{\text{old}} - \sqrt{W}A\delta U \,\|^2$$

(where $A := A(U_{\text{old}}, u_{\text{init}})$) becomes minimal. It is well known (see [2], pp. 109-119) that if A has maximal rank this minimisation problem has the unique solution

$$(23) \qquad \delta U \;=\; (A^T W A)^{-1} A^T W \rho_{\text{old}}\,.$$

However, the matrix $A^T W A$ is often ill-conditioned; thus for numerical reasons it is not recommended to perform the matrix inversion in (23) in a straightforward way. Instead, we determine an orthogonal matrix P such that

$$(24) \qquad P\sqrt{W}A \;=:\; R \;=\; \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

has upper triangular form (where R_1 is an upper triangular square matrix whose size is given by the number of solve-for parameters). (Such a matrix P can be determined by a sequence of *Householder transformations*; see [1], pp. 57-67, and [4], pp. 164-168.) We let

$$(25) \qquad \xi := P\sqrt{W}\rho_{\text{old}};$$

since applying an orthogonal matrix does not effect the norm of a vector, (22) becomes

$$(26) \qquad \begin{aligned} Q_{\text{new}} &= \|\xi - R\,\delta U\|^2 \;=\; \left\| \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} - \begin{bmatrix} R_1 \delta U \\ 0 \end{bmatrix} \right\|^2 \\ &= \|\xi_1 - R_1 \delta U\|^2 + \|\xi_2\|^2; \end{aligned}$$

it is clear that this last expression is minimised by letting

$$(27) \qquad \delta U \;:=\; R_1^{-1}\xi_1.$$

Note that (27) yields (23) because

$$(28) \qquad \begin{aligned} & R_1^{-1}\xi_1 \;=\; (R_1^T R_1)^{-1} R_1^T \xi_1 \;=\; (R^T R)^{-1} R^T \xi \;= \\ & (A^T \sqrt{W}^T P^T P \sqrt{W} A)^{-1} A^T \sqrt{W}^T P^T P \sqrt{W} \rho_{\text{old}} \end{aligned}$$

which, using $P^T P = \mathbf{1}$, becomes $(A^T W A)^{-1} A^T W \rho_{\text{old}}$. Thus we know how the update step (15) should be performed. Since in each step we linearised about the current estimate, iteration of the procedure is necessary. To monitor convergence, we note from (20) that we can predict which residual vector can be expected in the next iteration (to be performed with U_{new} instead of U_{old}), namely

$$(29) \qquad \rho_{\text{expected}} = \rho_{\text{old}} - A(U_{\text{old}}, u_{\text{init}})\delta U .$$

We consider convergence to be achieved if the difference between the residual vectors expected for and actually obtained in the next iteration becomes "small"; i.e., if

$$(30) \qquad \max_{1 \le i \le N} \left| \left(\sqrt{W}(\rho_{\text{expected}} - \rho_{\text{obtained}}) \right)_i \right| < \varepsilon$$

for some predefined convergence margin $\varepsilon > 0$. It remains to assess the accuracy of the estimate obtained. After convergence, all remaining residuals are supposed to stem exclusively from the measurement noise and the uncertainty in the consider parameter estimate u_{init} (whereas the final estimate obtained for U is supposed to be the true value $U^\star$). Then, if $\delta u := u_{\text{init}} - u^\star$ is the error in the estimate for u, the residual vector becomes

$$(31) \qquad \begin{aligned} \rho &= \widehat{\mu} - \mu(U^\star, u^\star + \delta u) \\ &= \mu(U^\star, u^\star) + n - \mu(U^\star, u^\star + \delta u) \\ &\approx \mu(U^\star, u^\star) + n - \mu(U^\star, u^\star) - (\partial\mu/\partial u)(U^\star, u^\star)\delta u \\ &= n - (\partial\mu/\partial u)(U^\star, u^\star)\delta u \\ &\approx n - (\partial\mu/\partial u)(U^\star, u_{\text{init}})\delta u . \end{aligned}$$

Making the (natural) assumption that the measurement noise and the error in the consider parameter estimate are uncorrelated and writing

$$(32) \qquad B := \frac{\partial \mu}{\partial u}(U^\star, u_{\text{init}}),$$

we find from (31) that

$$(33) \qquad \begin{aligned} \text{Cov}[\rho] &= \text{Cov}[n] + \text{Cov}[B\,\delta u] \\ &= W^{-1} + B\,\text{Cov}[\delta u] B^T \end{aligned}$$

and hence from (23) that

$$(34) \qquad \begin{aligned} \text{Cov}[\delta U] &= (A^T W A)^{-1} A^T W\, \text{Cov}[\rho] W A (A^T W A)^{-1} \\ &= (A^T W A)^{-1} + D\,\text{Cov}[\delta u] D^T \end{aligned}$$

where $D := (A^T W A)^{-1}(A^T W B)$. Note that the first summand in (34) represents the parameter estimation inaccuracy due to the noise in the measurements whereas the second summand represents the parameter estimation inaccuracy due the consider parameter uncertainty.

5. Nonlinear Update

When the general estimation procedure explained in the previous section is applied to the calibration problem at hand, a peculiarity arises. Namely, the parameter g (and hence the solve-for parameter $U = (p, g)$) cannot be taken as an element of a linear manifold, due to the constraint $g \in \text{SO}(3)$. As stated at the end of section 3, this is dealt with by allowing only updates of the special form (10). This has the consequence that two typical rows of the partial derivative matrix (21), associated with the u-coordinate und the v-coordinate of the same image point obtained as a measurement, take the forms

$$(35) \qquad \begin{bmatrix} (\nabla_p U)^T & (\widetilde{\nabla_g U})^T \\ (\nabla_p V)^T & (\widetilde{\nabla_g V})^T \end{bmatrix}$$

where

$$(36) \qquad \begin{aligned} \nabla_p U &= f \cdot \frac{(p - x) \times g_2}{\langle p - x, g_3 \rangle^2} = f \cdot \frac{g(y \times e_2)}{\langle y, e_3 \rangle^2}, \\ \nabla_p V &= -f \cdot \frac{(p - x) \times g_1}{\langle p - x, g_3 \rangle^2} = -f \cdot \frac{g(y \times e_1)}{\langle y, e_3 \rangle^2}, \\ \widetilde{\nabla_g U} &= \frac{-f}{\langle y, e_3 \rangle^2}\, y \times (y \times e_2), \\ \widetilde{\nabla_g V} &= \frac{f}{\langle y, e_3 \rangle^2}\, y \times (y \times e_1) \end{aligned}$$

according to (4), (5), (12) and (13); the update vector then takes the form

$$(37) \qquad (\delta p_1,\ \delta p_2,\ \delta p_3,\ \Delta_1,\ \Delta_2,\ \Delta_3)^T .$$

Once the update vector is found, we can form δg via (10), but, of course, we cannot simply write $g_{\text{new}} = g_{\text{old}} + \delta g$ because the right-hand side of this equation is not an element of SO(3). What we do instead is "wrap around" the vector δg and hence apply the update along the geodesic of SO(3) originating from g_{old} determined by the tangent vector δg. This results in the nonlinear update step

$$(38) \qquad g_{\text{new}} := g_{\text{old}} \exp\big(L(\Delta)\big)$$

where $\exp : \text{so}(3) \to \text{SO}(3)$ is the exponential function of the Lie group SO(3) which is explicitly given by the Rodrigues formula

$$(39) \qquad \begin{aligned} \exp\big(L(\Delta)\big) &= \cos(\|\Delta\|)\mathbf{1} + \frac{\sin(\|\Delta\|)}{\|\Delta\|} L(\Delta) \\ &\quad + \frac{1 - \cos(\|\Delta\|)}{\|\Delta\|^2} \Delta \otimes \Delta \end{aligned}$$

where $\mathbf{1}$ denotes the (3×3) identity matrix. The nonlinear update step is shown pictorially in the following diagram.

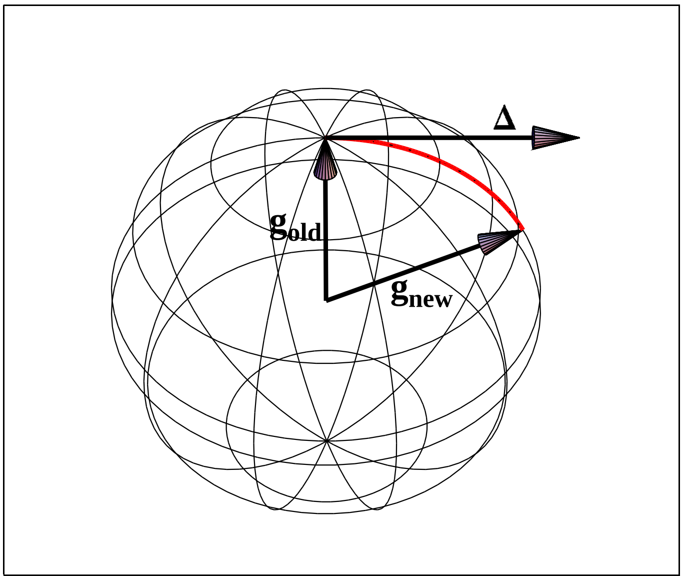

Figure 2: Nonlinear update of camera attitude.

After convergence, the $(6N \times 6N)$-covariance matrix of the parameter vector $(\delta p_1, \Delta_1, \ldots, \delta p_N, \Delta_N)$ can be computed using (34). To convert this into the covariance matrix for the state vector $(p_1, g_1, \ldots, p_N, g_N)$ we need to know how the covariance matrix of a random variable $\Delta \in \mathbb{R}^3$ is transformed into the covariance matrix of the random variable $g \exp\big(L(\Delta)\big)$. First we note that if $u, v, w \in \mathbb{R}^3$ then

$$(40)\qquad \begin{aligned} \big(L(u) \otimes L(v)\big)\big(L(w)\big) &= \langle\!\langle L(w), L(v) \rangle\!\rangle L(u) \\ &= 2\langle w, v\rangle L(u) \\ &= L\big(2\langle w, v\rangle u\big) \\ &= L\big(2(u \otimes v)(w)\big); \end{aligned}$$

hence if we identify $\mathbb{R}^3$ with so(3) via $\omega \mapsto L(\omega)$, then $L(u) \otimes L(v)$ is identified with $2\,u \otimes v$. Consequently,

$$(41)\qquad \begin{aligned} \mathrm{Cov}\big[L(\Delta)\big] &= E\big[L(\Delta) \otimes L(\Delta)\big] = E[2\,\Delta \otimes \Delta] \\ &= 2\,E[\Delta \otimes \Delta] = 2\ \mathrm{Cov}[\Delta]. \end{aligned}$$

By homogeneity of the Lie group SO(3), the accuracy of an estimate $g_{\text{new}} = g_{\text{old}} \exp\big(L(\Delta)\big)$ is the given by

$$(42)\qquad \mathrm{Cov}[g_{\text{new}}] = \mathrm{Cov}[L(\Delta)]\,.$$

6. Test Results

The algorithm described above was tested for a scenario representing an application in medical image processing as described in [3]. Four cameras are mounted at the corners of a rectangular frame.

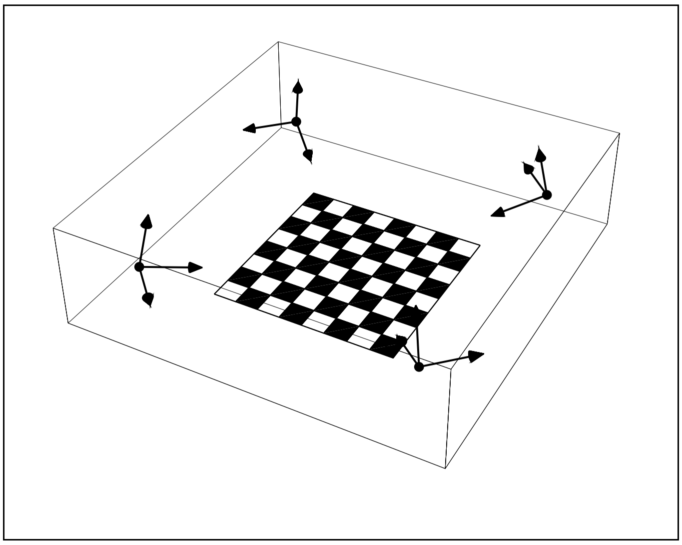

Figure 3: Scenario used for test runs.

Calibration is performed by taking pictures of landmarks arranged in a checkerboard pattern. We assume perfectly known landmarks and measurement accuracies of 0.01 units of length for both the u- and v-coordinates of all image points, where the unit length is taken as the length of a square in che checkerboard pattern used. (Test runs based on more realistic assumptions, using the landmark coordinates as consider parameters rather than known constants, will be included in the final version of the paper.) Rather large errors (half a unit length in position coordinates and ten degrees in angular coordinates) were introduced to obtain highly corrupted initial estimates for the camera positions and attitudes.

parameter	True values
p_1	(-2.000, -2.000, 2.000)
p_2	(-2.000, 10.000, 2.000)
p_3	(10.000, 10.000, 2.000)
p_4	(10.000, -2.000, 2.000)
g_1	0.162221 0.707107 0.688247 0.162221 -0.707107 0.688247 0.973329 0.000000 -0.229416
g_2	0.162221 -0.707107 0.688247 -0.162221 -0.707107 -0.688247 0.973329 0.000000 -0.229416
g_3	-0.162221 -0.707107 -0.688247 -0.162221 0.707107 -0.688247 0.973329 0.000000 -0.229416
g_4	-0.162221 0.707107 -0.688247 0.162221 0.707107 0.688247 0.973329 0.000000 -0.229416

Nevertheless, convergence to the correct solution (within the unavoidable error margins) was obtained within a few iterations; the estimation results obtained in the various iterations were as follows.

parameter	Initial estimates		
p_1	-1.07558	-2.74439	1.53538
p_2	-2.52006	10.21500	2.19312
p_3	10.93790	9.83020	1.80551
p_4	9.88609	-2.52936	1.87572
g_1	0.253780	0.761653	0.596222
	0.077723	-0.630466	0.772316
	0.964134	-0.149658	-0.219198
g_2	0.119402	-0.835456	0.536430
	-0.225317	-0.548999	-0.804880
	0.966941	-0.024762	-0.253794
g_3	-0.163257	-0.562758	-0.810339
	-0.169335	0.825154	-0.538931
	0.971943	0.049235	-0.230007
g_4	-0.330731	0.487284	-0.808190
	0.000936	0.856552	0.516060
	0.943725	0.169921	-0.283744

parameter	First iteration		
p_1	-1.98304	-2.22826	2.14633
p_2	-1.69773	9.75612	1.87365
p_3	9.80777	10.06730	2.00824
p_4	9.79324	-1.77262	1.93512
g_1	0.176914	0.711824	0.679712
	0.158173	-0.702191	0.694196
	0.971433	-0.015301	-0.236819
g_2	0.154134	-0.711256	0.685826
	-0.152950	-0.702930	-0.694620
	0.976140	0.002168	-0.217132
g_3	-0.155134	-0.717303	-0.679272
	-0.169122	0.696729	-0.697113
	0.973309	0.006734	-0.229398
g_4	-0.131674	0.696360	-0.705510
	0.138390	0.717650	0.682514
	0.981585	-0.007766	-0.190866

parameter	Second iteration		
p_1	-2.00022	-2.01701	1.98244
p_2	-1.97097	9.96164	2.02694
p_3	9.94319	10.02140	1.96921
p_4	9.97512	-1.99221	1.99222
g_1	0.163464	0.708437	0.686583
	0.158258	-0.705761	0.690547
	0.973773	-0.004222	-0.227482
g_2	0.162948	-0.706927	0.688260
	-0.167052	-0.707280	-0.686912
	0.972390	-0.003044	-0.233343
g_3	-0.157811	-0.710670	-0.685597
	-0.161018	0.703524	-0.692190
	0.974253	0.001158	-0.225454
g_4	-0.160756	0.707988	-0.687685
	0.163645	0.706222	0.688818
	0.973333	-0.001805	-0.229388

parameter	Third iteration		
p_1	-1.99356	-2.02457	1.98674
p_2	-1.99201	9.97217	2.03489
p_3	9.94417	10.01890	1.96908
p_4	9.99329	-2.01357	1.98964
g_1	0.163897	0.708827	0.686077
	0.158234	-0.705367	0.690956
	0.973704	-0.004685	-0.227768
g_2	0.162757	-0.706548	0.688695
	-0.167179	-0.707657	-0.686493
	0.972400	-0.003404	-0.233296
g_3	-0.157870	-0.710548	-0.685710
	-0.160931	0.703647	-0.692085
	0.974258	0.001093	-0.225434
g_4	-0.159845	0.708299	-0.687577
	0.163278	0.705910	0.689226
	0.973545	-0.002097	-0.228486

To monitor convergence, we list, for each iteration, the errors (i.e., deviations between currently best estimates and true values) for the camera positions and attitudes and also the root of the mean square error (RMS) as an indicator of the overall size of the residual vector. (The size of the attitude error is measured with respect to the matrix norm $\|A\| := \sqrt{\mathrm{tr}(A^T A)}$).

	initial	1st it.	2nd it.	3rd it.
p_1	1.274570	0.271662	0.024444	0.028651
p_2	0.594985	0.408422	0.055139	0.045341
p_3	0.972808	0.203820	0.068074	0.066565
p_4	0.555555	0.314103	0.027206	0.018347
g_1	0.249989	0.026066	0.007118	0.007845
g_2	0.291866	0.019900	0.007161	0.007501
g_3	0.272858	0.022672	0.009322	0.009138
g_4	0.449222	0.060607	0.003104	0.004042
RMS	379.035	57.2963	24.3506	24.2789

7. References

[1] Gerald J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York 1977

[2] Theodore D. Moyer, *Mathematical Formulation of the Double-Precision Orbit Determination Program (DPOPD)*, Technical Report 32-1527, Jet Propulsion Laboratory, Pasadena 1971

[3] D. Richter, F. La Torre, J. Egger, G. Straßmann, *Tetraoptical Camera System for Medical Navigation*, Proc. 17th Biennial International Eurasip Conference Biosignal 2004

[4] Josef Stoer, *Einführung in die Numerische Mathematik I*, Springer-Verlag, Berlin – Heidelberg – New York ^{3}1979