Principled Bundle Block Adjustment with Multi-head Cameras

Eleonora Maset[a], Luca Magri[b], Andrea Fusiello[a,*]

[a] *Dipartimento Politecnico di Ingegneria e Architettura, Universita degli Studi di Udine, Via Delle Scienze, 208, Udine, 33100, Italy*
[b] *Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, Milano, 20133, Italy,*

Abstract

This paper examines the effects of implementing relative orientation constraints on bundle adjustment, as well as provides a full derivation of the Jacobian matrix for such an adjustment, that can be used to facilitate other implementations of bundle adjustment with constrained cameras. We present empirical evidence demonstrating improved accuracy and reduced computational load when these constraints are imposed.

Keywords: Bundle Adjustment, Orientation, Oblique images, Multi-head camera, Multi-camera

1. Introduction

The use of oblique imaging technology is becoming increasingly popular in the geospatial industry, opening up a wide range of new possibilities for geospatial companies (Remondino and Gerke, 2015; Toschi et al., 2019). The technology provides high-resolution, three-dimensional perception of buildings, roads, and other urban objects from multiple angles. This allows for more detailed analysis and more accurate identification of objects and structures that are difficult to view from a straight-down perspective, such as tall buildings, bridges, and other large structures (Haala and Rothermel, 2015). The oblique angle ensures more detailed views of the object, which can be used for accurate measurements and mapping. Moreover, oblique photogrammetry represents a key technology for 3D city modelling, especially for roofs and facades reconstruction and texturing (Wang et al., 2023).

However, this technology also poses significant challenges. Oblique imaging requires dealing with large variations in image scale and illumination, multiple occlusions and a larger disparity search space. This means that the data captured by oblique imaging must be processed and analyzed more carefully, which can be a time-consuming and complex process. With regard to image block orientation, the main problem is how to efficiently and rigorously handle a larger number of unknowns while minimizing the risk of divergence in bundle block adjustment (BBA).

One of the first approaches developed to deal with a multi-head camera system was implemented for the post-processing of four panchromatic images simultaneously acquired by the Digital Mapping Camera (DMC) by Z/I Imaging Corporation (Madani et al., 2004). In that case, the relative position of the projection centers were precisely known from a calibration process, whereas relative orientation angles were estimated via BBA for each image exposure, thus handling possible angular instability over short time spans. Finally, the relative exterior orientation parameters were applied to generate a single, virtual perspective image for subsequent standard photogrammetric processes (Dörstel et al., 2002). Several works in the literature suggest considering relative orientation constraints between cameras (Wiedemann and Moré, 2012; Rupnik et al., 2015; Sun et al., 2016). In terms of available implementations, two main approaches have been proposed to handle rigid orientation constraints: the first, frequently applied in commercial surveys with ground and airborne multi-camera systems, consists of retrieving relative orientations between cameras during a calibration procedure (Esquivel et al., 2007; Dai et al., 2009; Schneider and Förstner, 2013). After this initial phase, BBA will optimise only the exterior orientation of a reference camera, while the others will be rigidly bound to it.

The second possible approach is to bypass the preliminary calibration and compute the relative orientations among the cameras directly from the data. This is done, with different nuances, in COLMAP (Schönberger and Frahm, 2016), and MicMac (Pierrot-Deseilligny et al., 2014). These methods start from an initialisation obtained from Structure-from-Motion (SfM) and deduce relative poses from the data, unlike other methods that derive relative poses from calibration. Also Pix4Dmapper (Pix4D) and Metashape (Agisoft) support the definition of multi-camera systems, but the available documentation does not give sufficient details on the underlying method.

The `RigBundleAdjuster` of COLMAP (Schönberger and Frahm, 2016) computes the average relative orientations between rigged cameras from the initial SfM and then

[*] Corresponding author
Email addresses: `eleonora.maset@uniud.it` (Eleonora Maset), `luca.magri@polimi.it` (Luca Magri), `andrea.fusiello@uniud.it` (Andrea Fusiello)

considers them known in the final BBA, as in a calibrated case (Heng et al., 2015). Note that they are introduced as hard constraints forced at each iteration, so the number of unknowns remain the same as in the unconstrained version of BBA.

The MicMac (Pierrot-Deseilligny et al., 2014) approach – in addition to being able to include pre-calibrated relative orientations – computes the initial relative orientations by averaging from the initial SfM. Then, it parametrises each unit of the multi-camera system with its independent extrinsic parameters and applies the relative orientations as soft constraints, thereby relaxing the rigidity of the rig. Similarly to (Schönberger and Frahm, 2016), this parametrisation does not reduce the number of BBA unknowns.

A third case, which we are not interested in, is when the rig is not considered rigid and the variable relative orientations are computed on-line. MicMac also enables this possibility with a weighting scheme that allows the relative orientation to remain constant or to evolve over time. In (Seok and Lim, 2019) the relative orientations can vary with the constraint that the distance between the cameras is constant.

This paper proposes a principled approach that introduces fixed relative orientations between rigged cameras as unknowns. We express the exterior orientation of the underlying cameras as a function of the reference camera parameters and the fixed, unknown relative orientations. In this way, the rigidity of the multi-camera system is respected while reducing the number of unknowns. In addition, this BBA with constrained relative orientation can replace the BBA which is routinely performed as the last stage of the SfM pipeline, whereas other approaches are implemented as an additional step *after* the unconstrained BBA.

A similar approach has been proposed by (Sun et al., 2016), but neither the implementation nor the details of the derivative calculation are available.

Our method, henceforth dubbed "Constrained Relative Orientations BBA" or CRO-BBA for short, implements the exact formulation of the Jacobian matrix, that collects the partial derivatives of the collinearity equations rewritten to account for relative constraints. The formulae for the Jacobian matrix of the BBA with constrained cameras have been explicitly derived using the "matrix differential calculus" formalism (Magnus and Neudecker, 1999), which leads to a compact and modular derivation.

We will provide evidence that exploiting the rigidity of the system actually has two advantages: it reduces the number of unknowns *and* it improves accuracy. In fact, neglecting the relative orientation constraint artificially increases the degrees of freedom of the system, leading to over-fitting (i.e. noise-fitting).

2. Notation and background

The central projection operated by a pinhole camera is represented mathematically by the collinearity equations:

$$\mathbf{x} = f(\mathbf{X}, \mathbf{g}) \tag{1}$$

where $\mathbf{X}$ are the coordinates of an object point, $\mathbf{x}$ are the two coordinates of the corresponding image point (in pixels) and $\mathbf{g}$ is the vector of extrinsic parameters that represents the exterior orientation of the image. The intrinsic parameters are assumed to be known and fixed, so they are subsumed in f. Their estimation, possibly including distortion parameters, can easily be added to the process.

The exterior orientation is a direct isometry (or rigid transform) that has a matrix representation (in homogeneous coordinates) as

$$G = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2}$$

where R is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 vector representing a translation. The rotation matrix, in turn, can be parameterized by three numbers, e.g, the Euler angles ω, φ, κ (see Appendix B), so we write $R = R(\omega, \varphi, \kappa)$ and the vector of extrinsic parameters is: $\mathbf{g} = [\omega; \varphi; \kappa; \mathbf{t}]$.

If the central projection is accomplished by:

$$f_P([x, y, z]^\top) = \left[\frac{x}{z}, \frac{y}{z}\right]^\top \tag{3}$$

then we can write the collinearity equations as:

$$\mathbf{x} = f(\mathbf{X}, \mathbf{g}) = f_P(R(\omega, \varphi, \kappa)\mathbf{X} + \mathbf{t}). \tag{4}$$

The center of projection (COP) is the single point in space where all the projection rays sampled by the pinhole camera intersect, and is given by $-R^\top t$.

3. Problem statement

Let us consider a multi-camera system (a.k.a. multi-head camera) composed by k cameras, where one is taken as the reference and the remaining $k-1$ have a fixed but unknown relative orientation with respect to the first one. For the sake of concreteness, one can think of the customary "Maltese cross" arrangement (Fig. 1), that comprises a single nadir-pointing camera and four oblique-pointing cameras ($k = 5$). Two of the oblique cameras point in opposite directions cross–track (left, right), while the remaining two oblique cameras point in opposite directions along-track (forward, backward). The five cameras are housed in a suitable frame that ensures rigidity and their shutters are synchronized to operate simultaneously[1].

[1] Small lags can be approximately absorbed into the relative orientation as translations along the direction of motion.

- Let I_ν be one nadir image, whose exterior orientation is

$$G_\nu = \begin{bmatrix} R_\nu & \mathbf{t}_\nu \\ \mathbf{0} & 1 \end{bmatrix}. \tag{5}$$

  G_ν transforms 3D point coordinates from an external (object) reference frame to the standard camera reference frame, whose origin is at the COP and the Z axis is orthogonal to the image plane;

- let

$$G_\tau = \begin{bmatrix} R_\tau & \mathbf{t}_\tau \\ \mathbf{0} & 1 \end{bmatrix} \tag{6}$$

  be the orientation of the oblique image relative to the nadir image, which are fixed but unknown. G_τ transform points from the nadir camera coordinate system to the oblique one; in the case of the "Maltese cross", there are four of such relative orientations: forward, backward, left, right, i.e., $\tau = \{F,B,L,R\}$ (see Fig. 1);

- let I_μ be one of the (four) oblique images relative to I_ν, whose exterior orientation is

$$G_\mu = \begin{bmatrix} R_\mu & \mathbf{t}_\mu \\ \mathbf{0} & 1 \end{bmatrix} \quad \mu = (\nu, \tau); \tag{7}$$

  then G_μ can be written as a function of the exterior orientation of the nadir image and the relative orientation:

$$G_\mu = G_\tau \cdot G_\nu = \begin{bmatrix} R_\tau \cdot R_\nu & R_\tau \cdot \mathbf{t}_\nu + \mathbf{t}_\tau \\ \mathbf{0} & 1 \end{bmatrix}; \tag{8}$$

- let us parametrize the nadir orientation G_ν with the 6 parameters $\mathbf{g}_\nu^\top = [\omega, \varphi, \kappa, \mathbf{t}_\nu]$, where the first three are the Euler angles;

- let us parametrize the relative orientation G_τ with the 6 parameters $\mathbf{g}_\tau^\top = [\theta, \psi, \varrho, \mathbf{t}_\tau]$ where the first three are the Euler angles.

Although we focused on a multi-camera system with oblique cameras, such as the "Maltese cross", CRO-BBA can be applied to any multi-camera system, such as, e.g., multi-spectral cameras that are composed by separate nadir-looking cameras that record specific light-wave bands.

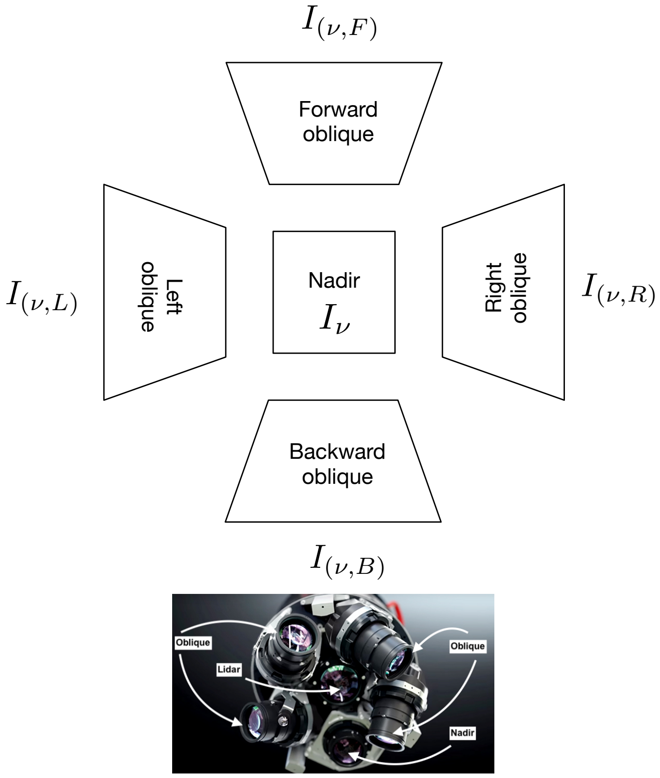


Figure 1: Left: illustration of the "Maltese cross" footprint. Right: an example of cameras arrangement in Leica CityMapper

4. Jacobian matrix

Performing BBA using any variant of the Gauss-Newton method (such as Levenberg-Marquardt or Gauss-Markov) entails computing the Jacobian matrix that collects the partial derivatives of f.

The primary structure of the Jacobian of the classical BBA is composed of blocks of two types, which we will call J_A and J_B, that contain respectively the derivatives of the collinearity equations with respect to (wrt) image orientation and wrt 3D point. In the case of CRO-BBA, only the nadir images give rise to this type of blocks, whereas oblique images (whose orientation is linked to the nadir image) lead to three new blocks (Fig. 2):

- J_C collects the derivative of the residual in the oblique images wrt the nadir orientation;
- J_D collects the derivative of the residual in the oblique images wrt the relative orientation;
- J_E collects the derivative of the residual in the oblique images wrt 3D point coordinates.

The blocks J_A, J_C and J_D have dimension 2×6, whereas J_B and J_E have dimension 2×3. We assume that the oblique images relative to the same nadir image are consecutive and ordered consistently.

Before working out the derivative of the function f defined in (4), let us establish the Jacobian of the projection function f_p by differentiating it element-by-element:

$$\mathcal{D} f_p(\mathbf{W}) = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix}, \quad \mathbf{W}^\top = [x, y, z]. \tag{9}$$

We now proceed to derive the expression for the non-zero blocks of the BBA Jacobian.

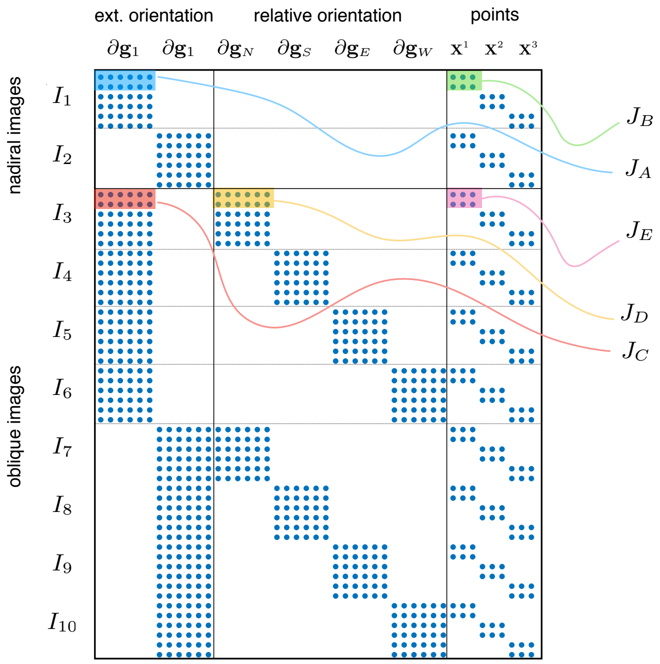


Figure 2: Primary structure of the CRO-BBA Jacobian for a block of 2 nadir cameras, 8 oblique and 3 tie-points

Instead of reverting to subscript notation for computing such derivatives, we perform the entire operation using the *matrix differential calculus* introduced by (Magnus and Neudecker, 1999), which we will briefly review in Appendix A.

Blocks J_A and J_B correspond to nadir images, and they are easily derived:

$$\begin{aligned} J_A = \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial g_\nu^\top} &= \frac{\partial f_p(\mathbf{W})}{\partial \mathbf{W}^\top}\bigg|_{\mathbf{W}=R_\nu\mathbf{X}+t_\nu} \frac{\partial(R_\nu\mathbf{X}+t_\nu)}{\partial g_\nu^\top} \\ &= \mathcal{D}f_p(R_\nu\mathbf{X}+t_\nu)\left[(\mathbf{X}^\top \otimes I_3)\mathcal{D}R_\nu | I_3\right] \end{aligned} \tag{10}$$

$$\begin{aligned} J_B = \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial \mathbf{X}^\top} &= \frac{\partial f_p(\mathbf{W})}{\partial \mathbf{W}^\top}\bigg|_{\mathbf{W}=R_\nu\mathbf{X}+t_\nu} \frac{\partial(R_\nu\mathbf{X}+t_\nu)}{\partial \mathbf{X}^\top} \\ &= \mathcal{D}f_p(R_\nu\mathbf{X}+\mathbf{t}_\nu)R_\nu \end{aligned} \tag{11}$$

As for the blocks related to oblique images, let us derive first with respect to the relative orientation

$$J_D = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\tau^\top} = \left[\frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\theta, \psi, \varrho]} \;\middle|\; \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\tau^\top} \right] \tag{12}$$

Let $\mathbf{X}' = R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau$, then

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\theta, \psi, \varrho]} &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau R_\nu\mathbf{X}+R_\tau\mathbf{t}_\nu+\mathbf{t}_\tau)}{\partial[\theta, \psi, \varrho]} \\ &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau(R_\nu\mathbf{X}+\mathbf{t}_\nu))}{\partial[\theta, \psi, \varrho]} \\ &= \mathcal{D}f_p(\mathbf{X}')\left((R_\nu\mathbf{X}+\mathbf{t}_\nu)^\top \otimes I_3\right)\mathcal{D}R_\tau \end{aligned} \tag{13}$$

and also

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\tau^\top} &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau R_\nu\mathbf{X}+R_\tau\mathbf{t}_\nu+\mathbf{t}_\tau)}{\partial \mathbf{t}_\tau^\top} \\ &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial \mathbf{t}_\tau}{\partial \mathbf{t}_\tau^\top} \\ &= \mathcal{D}f_p(\mathbf{X}')I_3 \end{aligned} \tag{14}$$

Now we derive with respect to the nadir orientation:

$$J_C = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\nu^\top} = \left[\frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\omega, \varphi, \kappa]} \;\middle|\; \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\nu^\top} \right] \tag{15}$$

where

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\omega, \varphi, \kappa]} &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau R_\nu\mathbf{X}+R_\tau\mathbf{t}_\nu+\mathbf{t}_\tau)}{\partial[\omega, \varphi, \kappa]} \\ &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial R_\tau R_\nu \mathbf{X}}{\partial[\omega, \varphi, \kappa]} \\ &= \mathcal{D}f_p(\mathbf{X}')(\mathbf{X}^\top \otimes R_\tau)\mathcal{D}R_\nu \end{aligned} \tag{16}$$

and

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\nu^\top} &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau R_\nu\mathbf{X}+R_\tau\mathbf{t}_\nu+\mathbf{t}_\tau)}{\partial \mathbf{t}_\nu^\top} \\ &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial R_\tau \mathbf{t}_\nu}{\partial \mathbf{t}_\nu^\top} \\ &= \mathcal{D}f_p(\mathbf{X}')R_\tau \end{aligned} \tag{17}$$

Finally, for the 3D point:

$$\begin{aligned} J_E = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{X}^\top} &= \mathcal{D}f_p(\mathbf{X}')\frac{\partial(R_\tau R_\nu\mathbf{X}+R_\tau\mathbf{t}_\nu+\mathbf{t}_\tau)}{\partial \mathbf{X}^\top} \\ &= \mathcal{D}f_p(\mathbf{X}')R_\tau R_\nu \end{aligned} \tag{18}$$

It is worth noting that these formulae are modular with respect to the parametrization chosen for R: changing it will cause only $\mathcal{D}R$ to change. Intrinsic parameters may possibly be added to the unknowns.

Using these formulae, we implemented CRO-BBA in MATLAB adopting camera reduction (Brown, 1958) and the Levenberg-Marquardt optimization strategy. Levenberg-Marquardt is a variation of Gauss-Newton, where a diagonal term is added to the Hessian matrix. By changing the weight of this term the algorithm can dynamically move between pure Gauss-Newton and gradient descent, improving convergence basin and speed (Börlin and Grussenmeyer, 2013). Moreover, this diagonal matrix (a.k.a. damping term) has a regularization effect that restores the full rank of the Jacobian matrix, thereby implicitly removing the datum defect (Triggs et al., 2000) of the free-network adjustment. Classical BBA is obtained by switching off relative orientation constraints in our MATLAB implementation.

5. Experiments

We performed tests on simulated data and one real data set.

Let m be the number of nadir images and let k be the number of sensors that compose the multi-camera system (the total number of images is mk), n be the number of tie-points, ℓ be the number of equations and p be the number of unknowns. If all tie-points were seen in all images $\ell = 2mkn$, but in practice it is much smaller than that, for it depends on the visibility. As for p, if the oblique images are adjusted as if their orientations were independent, the system has $p = 6mk + 3n$ unknowns, whereas by enforcing the rigidity, the unknowns reduce to $p = 6(m + k - 1) + 3n$. This means that CRO-BBA solves a significantly smaller system of equations, with benefits in computing time, memory footprint and stronger block stability.

As image-space errors we considered the RMS (Root Mean Square) reprojection error and the root of the reference variance (RRV) defined as follows.

The reprojection error is the $\ell \times 1$ vector $\mathbf{v}$ whose entries are given by $\mathbf{x}_{ij} - f(\mathbf{X}_i, g_j)$ for $i = 1 \ldots n, \quad j = 1 \ldots mk$. The RMS reprojection error - RMSRE is given by:

$$RMSRE = \sqrt{\frac{\mathbf{v}^\top \mathbf{v}}{\ell}}, \tag{19}$$

while the reference variance - RRV (Kraus, 2007) is computed as:

$$RRV = \sqrt{\frac{\mathbf{v}^\top \mathbf{v}}{\ell - p}}. \tag{20}$$

Our BBA/CRO-BBA do not use control points, producing a free-network solutions. Then we align this result to the datum by computing the similarity (a.k.a. S-transform, or Helmert transform) that minimizes the least-squares distances between corresponding object points and control points. Thus, in object-space, the evaluation considered the RMS of the residual distances of this alignment. Any non-Euclidean deformation of the model generated by BBA/CRO-BBA is revealed (in object-space) by the alignment residuals.

In the simulated experiments the COPs are known as well, so we also (separately) aligned the reconstructed COPs to their known positions and measured the RMS of the residual distances[2].

5.1. Simulated data

Simulation used randomly subsampled 3D points and trajectory from a real LiDAR (Light Detection And Ranging) survey of the city of Udine, Italy (data courtesy by Helica s.r.l.). We assumed a multi-camera system composed of five Phase One iXA 180 (10,328 × 7,760 pixels, 50 mm focal length) in a "Maltese cross" arrangement. Angles between the principal axis of the nadir camera and those of the oblique images were set to 30°, whereas the distances between the center of the nadir and the oblique cameras were 0.20 m. Altitude and shooting frequency were adapted so as to obtain a lateral overlap (cross-track between nadir images) of 30%, while the forward overlap (along-track between nadir images) was set to 60%. As a result, this block consists of 400 images (80 nadir and 320 oblique) and 700 tie-points (Fig. 3), with an average of 30 points visible in each image and a mean GSD (Ground Sampling Distance) of 6 cm/pixel in the nadir images.

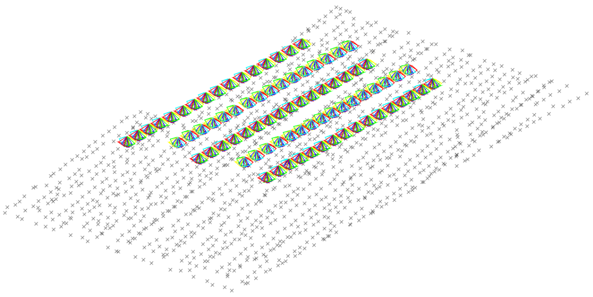


Figure 3: Simulated data. Cameras are represented by coloured pyramids, control/tie-points points are displayed as gray crosses (best viewed in color)

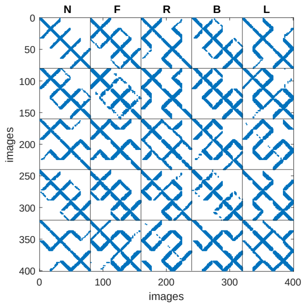


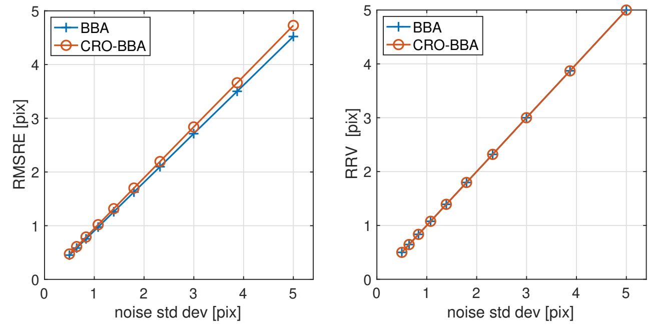
Figure 4: Simulated data. Matches distribution between the images. Images are sorted according to the ordering "Nadir", "Forward", "Right", "Backward" and "Left" cameras.

Figure 4 shows the distribution of matches, which configures a tightly bound block, more than what found in a real scenario (like in Fig. 11).

To carry out the simulation, we defined the ground truth exterior orientations on the basis of the flight plan described above. Then we projected the object points on the image planes adding different values of random Gaussian noise to the image coordinates, choosing as standard deviations 10 logarithmically spaced values from 0.5 to 5.0 pixels.

The initial values of the exterior orientation of nadir images were obtained by perturbing the ground truth with random Gaussian noise with $\sigma = 0.20$ m for the position and $\sigma = 0.2°$ for the Euler angles. Moreover, we added a

[2] Actually they are known also in the real experiment from the onboard GNSS/INS, but their accuracy is not sufficient to qualify them as reference values, for the purposes of this study.



(a)

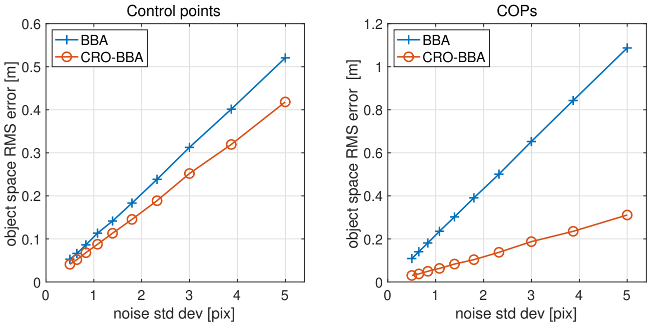
(b)

Figure 5: Image-space errors: RMS of the residuals (RMSRE) and Root of Reference Variance (RRV) for BBA and CRO-BBA vs image noise. RRV is equal to the standard deviation of the image noise

random noise with $\sigma = 0.05$ m and $\sigma = 0.05°$ to the relative positions and angles of the oblique cameras, respectively. For each setting the trial was run 100 times.

Figure 5(a) shows that the RMSRE (averaged over 100 trials) is slightly larger for CRO-BBA than BBA. This is in line with the results shown in (Sun et al., 2016), on the basis of which the authors conclude that adding constraints worsen the accuracy of BBA.

However, looking at the reprojection error without considering the degrees of freedom of the model to be fitted leads to biased conclusions. In fact, unconstrained BBA has more degrees of freedom (d.o.f.) than CRO-BBA, being thus more capable of reducing the residuals by fitting the noise. Indeed, when the d.o.f. of the model are taken into account by computing the RRV (Fig. 5(b)), this is practically the same and it is also equal to the standard deviation of the noise added to the image points, as one should expect.



(a)

(b)

Figure 6: Object-space errors: RMS of the alignment residuals computed on control points (a) and COPs (b) vs image noise

The advantages of CRO-BBA can be appreciated in object-space (Fig.s 6(a) and 7), by computing the alignment residuals wrt the control points (all tie-points are also control points in this simulation). These plots clearly show (with a confidence close to 100% according with the t-test) that the unconstrained BBA produces a less accurate estimate of object points *and* COPs. The specific distribution of these errors depends on the arbitrary choice of the datum, but the qualitative observation that unconstrained BBA is less accurate is independent.

In a practical scenario, these residuary non-Euclidean deformations would have been compensated for by ground control points, which is why this effect is rarely detected: the use of a model with an excess number of d.o.f. is balanced by additional constraints in the form of ground control points.

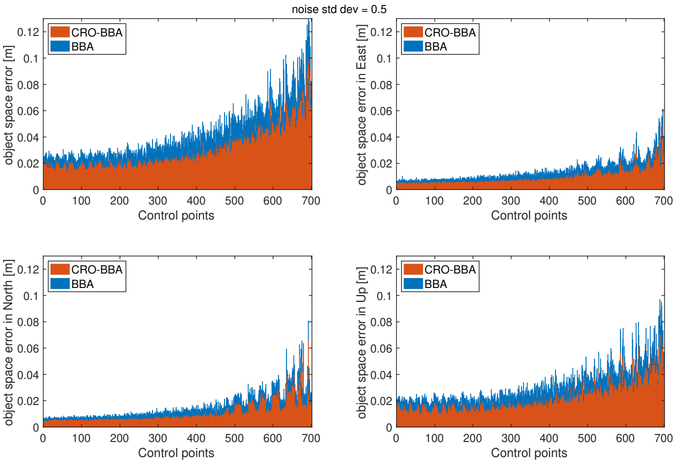


Figure 7: Object-space errors for control points: norm of the error and ENU components. Added noise has 0.5 pixels standard deviation. Points are sorted by decreasing ray-multiplicity. This figure is best viewed in colour

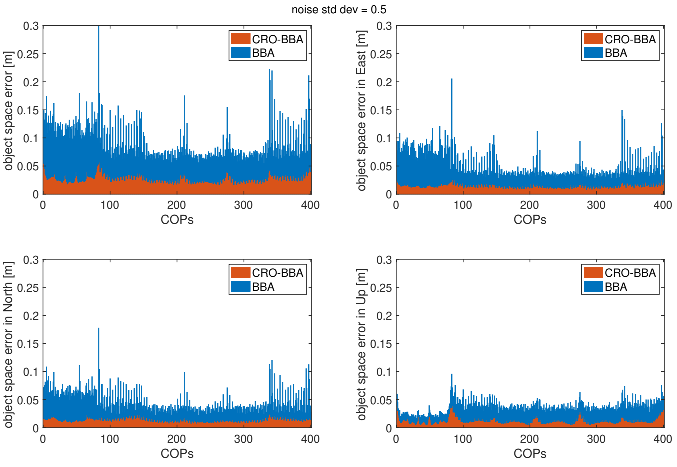
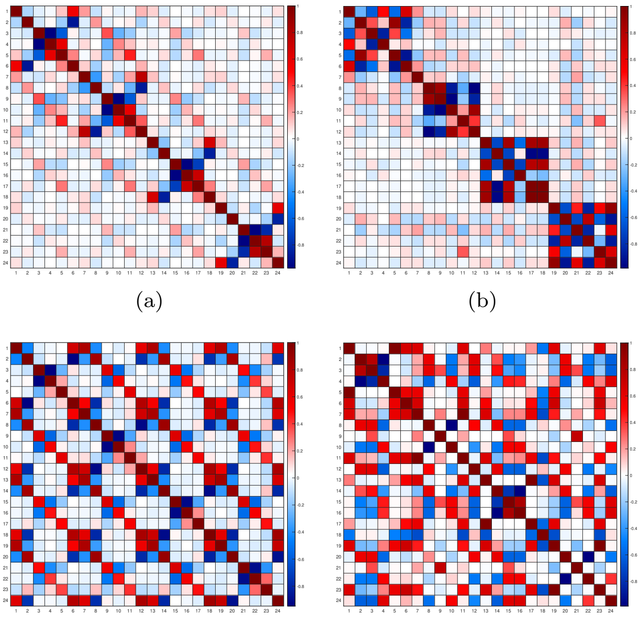


Figure 8: Object-space errors for COPs: norm of the error and ENU components. Added noise has 0.5 pixels standard deviation. This figure is best viewed in colour

It is worth noting also that while for CRO-BBA the errors affecting control points and COPs are comparable (please note the different scales of y-axes for Fig. 6(a) and Fig. 6(b)), BBA produces a worst localization of the COPs.

This is again dependent of the datum, but since we set it implicitly on control points when evaluating the error on control points and on COPs when evaluating the error on COPs, we conclude that it reflects an intrinsically worst



(a)

(b)

(c)

(d)

Figure 9: Top: correlation among the parameters (6 for each camera) of the first 4 nadir cameras (a) and the 4 oblique cameras (b) in unconstrained BBA. Bottom: correlation among the parameters of the first 4 nadir cameras (c) and the 24 relative orientation parameters for CRO-BBA (d)

localization of COPs in BBA. This can be explained in terms of the degrees of freedom of the system: in BBA the COPs of the oblique cameras are independent variables, while the imposition of relative constraint in CRO-BBA reduces the degrees of freedom and the error.

In particular, the nadir images (they correspond to the first 80 COPs in Fig. 8) in BBA have an average error (0.123 m) which is significantly higher (with nearly 100% confidence according to the t-test) than the oblique ones (0.085 m). So, ignoring relative orientation constraints affects the localization of the nadir images more than the oblique images.

Fig. 9 reports the correlation among the unknown parameters in the classical BBA and CRO-BBA scenario, for a subset of cameras. It can be noticed that in classical BBA the correlation for nadir and oblique cameras is fairly similar and has a diagonal block structure, evidence of a weaker block geometry. In the case of CRO-BBA, the nadir cameras are much more correlated, as expected, and so are the 24 relative orientation parameters that link the oblique cameras to the nadir ones (Fig. 9(d)).

5.2. Real data from Leica CityMapper

Real data were collected by the Leica CityMapper (1st generation) hybrid sensor over the city of Heilbronn, Germany (data courtesy by Leica Geosystems). Leica CityMapper (Fig. 1) combines a Hyperion LiDAR unit (1064 nm wavelength, theoretical ranging accuracy <2 cm) and a multi-camera system, featuring one nadir-looking camera head (RGB CCD size 10,320 x 7,752 pixels, NIR CCD size 3,336 x 4,500 pixels, 83 mm focal length) and four 45°-tilted camera heads (RGB CCD size 10,320 x 7,752 pixels, 156 mm focal length).

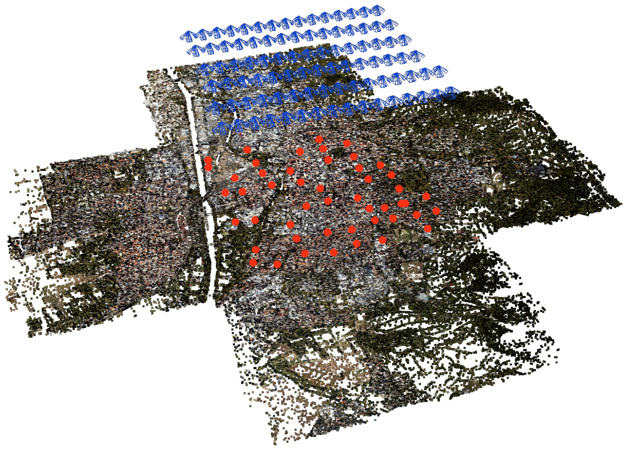


Figure 10: Real data used for the experiment. Cameras are represented by blue pyramids, tie-points has their own natural color, control points are displayed as red dots (best viewed in color)

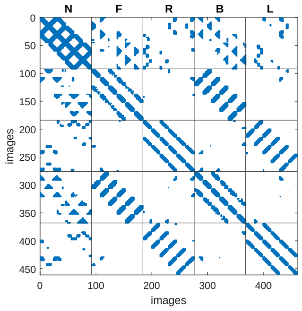


Figure 11: Matches distribution between the images. Images are sorted according to the ordering “Nadir”, “Forward”, “Right”, “Backward” and “Left” cameras.

The flight plan was designed using an average nadir GSD of 12 cm, and along-across overlaps of 80% and 60%, respectively. The selected subset includes 460 images (92 nadir images and 368 oblique images) and covers an area of ca. 3.5 km x 3.5 km over the city centre (Fig. 10). As ground truth data we employed 49 control points, surveyed with RTK GNSS with a mean 3D accuracy of 5 cm.

In the experiments with real data we used 3DF Zephyr to extract and match SIFT-like features and fed these image correspondences to our BBA/CRO-BBA. From the matches distribution shown in Fig. 11, it can be noticed that correspondences mainly exist between cameras pointing in the same absolute direction, corresponding to the diagonal blocks (along and cross-track) plus the forward-backward and right-left ones (only cross-track). While the nadir camera has some matches with the others, the other blocks are mostly empty.

For the intrinsic parameters, we adopted the values reported in the calibration certificate of the system. Then we computed the errors as in the previous cases, with the difference that the COPs are unknown and that the (49) control points are only a subset of the (8049) tie-points. The RRV is 1.021 pixel for BBA and 1.043 pixel CRO-BBA, while the object-space errors wrt control points are shown in Fig. 12, where it can be appreciated that the CRO-BBA produces more accurate results than BBA. The RMS of the object-space errors is 0.209 m for BBA and 0.153 m for CRO-BBA, and the t-test confirms that the two distributions are indeed different with 99.47% confidence. This experiment is in agreement with previous simulations: the enforcement of relative constraints is advantageous in terms of error in object-space, although the error in image-space (RRV) is approximately the same.

Figure 13 is a box-plot of the statistics of the offsets from the COP of one oblique image to the COP of the corresponding nadir image, after BBA and after CRO-BBA (in the latter case the values are constant). As we do not know the calibrated reference values, this can be taken

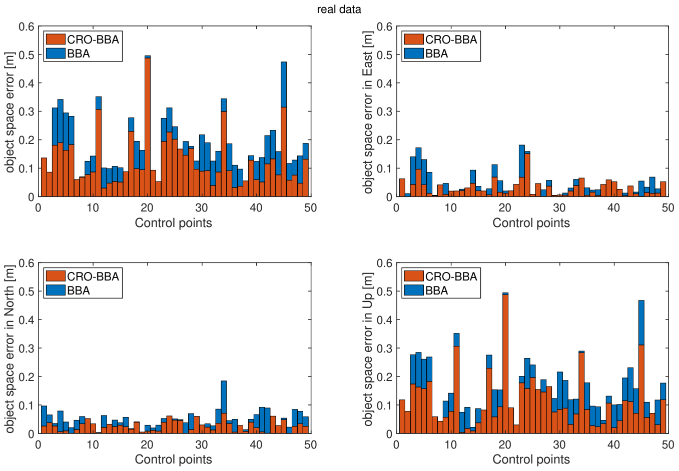


Figure 12: Results of the experiment with real data. Object-space errors for control points: norm of the error and ENU components. This figure is best viewed in colour

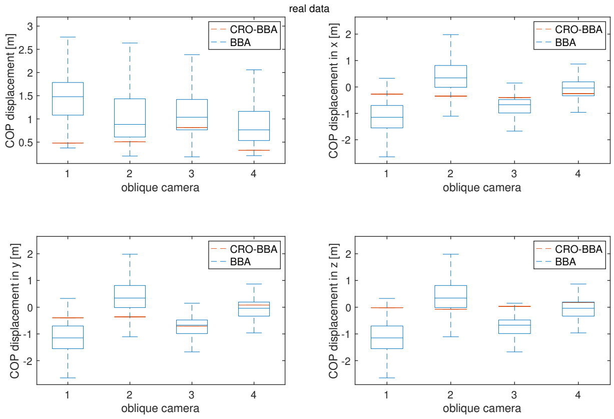


Figure 13: Results of the experiment with real data. Box plots of the oblique COP offsets: norm of the error ad components. This figure is best viewed in colour

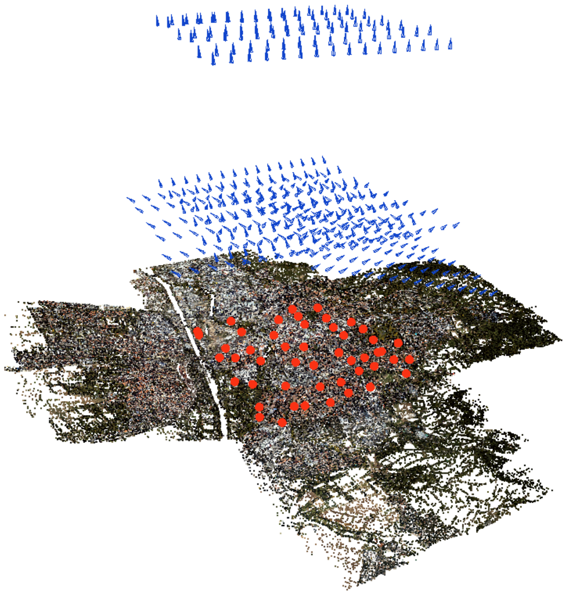


Figure 14: When intrinsic parameters are left free, the nadir images are localized with a large altitude error in unconstrained BBA

as an indicator of the precision of the COP localization achieved by BBA. Please note that the scale of y-axis is in meters.

Although all these experiments have been done with fixed interior orientation (our MATLAB implementation only works with fixed intrinsic parameters), it is instructive to see what would happen leaving them free in an unconstrained BBA (with 3DF Zephyr). The result is shown in Fig. 14, where it is noticeable that the nadir cameras had been offset from the corresponding oblique ones. This is due to the correlation between height of the COP and focal length, which gets stronger as the perspective effect is weaker, as in the nadir views, and so these views are more affected (see, e.g., (Luhmann et al., 2016)). The effect disappears if interior orientation is fixed, although, as we showed, the COPs of nadir images still suffer from a larger error than the oblique ones if relative orientation constraints are dropped.

6. Conclusions

In this paper we studied bundle adjustment with constrained cameras. This means that the orientation of certain cameras is expressed in relation to others, and these relative orientations become part of the unknowns. Though some have argued that this imposition of constraints decreases the accuracy of the results, we have found the opposite to be true. By capitalizing on the system's rigidity, the number of unknowns decreases and accuracy improves. Our experiments with simulated and real data have shown that there is no significant difference in errors in image space between solutions with and without relative orientation constraints, while errors in object space demonstrate that the unconstrained solution is severely deformed, reducing the accuracy of the result. All in all, there is no reason to avoid using relative orientation constraints in the bundle adjustment of image blocks in multi-camera systems.

Acknowledgments

Luca Magri has been supported by “HEaD Higher Education and Development Project” FP1619942003 and by 3Dflow s.r.l. Thanks to Helica s.r.l. for providing the flight data for the simulations, and to Leica Geosystems for providing the CityMapper dataset. Isabella Toschi coauthored a preliminary conference version of this paper (Maset et al., 2020). The authors are grateful to Fabio Crosilla and the anonymous reviewers for their valuable comments that improved the quality of the paper.

References

Brown, D., 1958. A solution to the general problem of multiple station analytical stereotriangulation. Technical Report AFMTC TR 58-8. Patrick Airforce Base, Florida.

Börlin, N., Grussenmeyer, P., 2013. Bundle adjustment with and without damping. The Photogrammetric Record 28, 396–415.

Dai, Y., Trumpf, J., Li, H., Barnes, N., Hartley, R., 2009. Rotation averaging with application to camera-rig calibration, in: Asian Conference on Computer Vision, Springer. pp. 335–346.

Dörstel, C., Zeitler, W., Jacobsen, K., 2002. Geometric calibration of the dmc: method and results. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 1, 324–333.

Esquivel, S., Woelk, F., Koch, R., 2007. Calibration of a multi-camera rig from non-overlapping views, in: Joint Pattern Recognition Symposium, Springer. pp. 82–91.

Haala, N., Rothermel, M., 2015. Image-based 3d data capture in urban scenarios, in: Proc. Photogrammetric Week 2015, pp. 119–130.

Heng, L., Furgale, P., Pollefeys, M., 2015. Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. Journal of Field Robotics 32, 775 – 802.

Kraus, K., 2007. Photogrammetry - Geometry from Images and Laser Scans - 2nd edition. Walter de Gruyter.

Lucas, J.R., 1963. Differentiation of the orientation matrix by matrix multipliers. Photogrammetric Engineering 29, 708–715.

Luhmann, T., Fraser, C., Maas, H.G., 2016. Sensor modelling and camera calibration for close-range photogrammetry. ISPRS Journal of Photogrammetry and Remote Sensing 115, 37 – 46.

Madani, M., Dörstel, C., Heipke, C., Jacobsen, K., 2004. Dmc practical experience and accuracy assessment. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 35, 396–401.

Magnus, J.R., Neudecker, H., 1999. ”Matrix Differential Calculus with Applications in Statistics and Econometrics”. Revised ed., John Wiley & Sons.

Maset, E., Magri, L., Toschi, I., Fusiello, A., 2020. Bundle block adjustment with constrained relative orientations. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2020, 49–55.

Pierrot-Deseilligny, M., Jouin, D., Belvaux, J., Maillet, G., Girod, L., Rupnik, E., Muller, J., Daakir, M., Choqueux, G., Deveau, M., 2014. Micmac, apero, pastis and other beverages in a nutshell. Institut Géographique National .

Remondino, F., Gerke, M., 2015. Oblique aerial imagery – a review, in: Proc. Photogrammetric Week 2015, pp. 75–83.

Rupnik, E., Nex, F., Toschi, I., Remondino, F., 2015. Aerial multi-camera systems: Accuracy and block triangulation issues. ISPRS Journal of Photogrammetry and Remote Sensing 101, 233 – 246.

Schneider, J., Förstner, W., 2013. Bundle adjustment and system calibration with points at infinity for omnidirectional camera systems. Photogrammetrie-Fernerkundung-Geoinformation 2013, 309–321.

Schönberger, J.L., Frahm, J.M., 2016. Structure-from-Motion Revisited, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Seok, H., Lim, J., 2019. Rovo: Robust omnidirectional visual odometry for wide-baseline wide-fov camera systems, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE Press. p. 6344–6350.

Sun, Y., Sun, H., Yan, L., Fan, S., Chen, R., 2016. Rba: Reduced bundle adjustment for oblique aerial photogrammetry. ISPRS Journal of Photogrammetry and Remote Sensing 121, 128–142.

Toschi, I., Remondino, F., Hauck, T., Wenzel, K., 2019. When photogrammetry meets lidar: Towards the airborne hybrid era. GIM International Sept./Oct.

Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 2000. Bundle adjustment - a modern synthesis, in: Proceedings of the International Workshop on Vision Algorithms, Springer-Verlag. pp. 298–372.

Wang, F., Zhou, G., Hu, H., Wang, Y., Fu, B., Li, S., Xie, J., 2023. Reconstruction of lod-2 building models guided by façade structures from oblique photogrammetric point cloud. Remote Sensing 15, 400.

Wiedemann, A., Moré, J., 2012. Orientation strategies for aerial oblique images. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 39, 185–189.

Appendix A. Matrix differential calculus

While for a differentiable function $\mathbb{R}^n \to \mathbb{R}^m$ the partial derivatives are customary arranged into a matrix called the Jacobian, for matrix functions $\mathbb{R}^{n\times q} \to \mathbb{R}^{m\times p}$ naturally raises the question of how to pack the *mnpq* partial derivatives, because this can be done in many ways. The following notation was introduced by (Magnus and Neudecker, 1999), who argue that it should be adopted for a number of good reasons, the most important of which is that it allows the use of the chain rule.

Definition 1. *Let F be a differentiable $m \times p$ real matrix function of a $n \times q$ matrix of real variables X. The Jacobian matrix of F at X is the $mp \times nq$ matrix*

$$\mathcal{D}F(X) = \frac{\partial \operatorname{vec} F(X)}{\partial (\operatorname{vec} X)^\top}. \tag{A.1}$$

where $\operatorname{vec}(A)$ is the column vector obtained by stacking the columns of A. Note that (A.1) also defines the Jacobian matrix for vector functions of vector variables.

Definition 1 reduces the study of matrix functions of matrices to the study of vector functions of vectors, since it allows $F(X)$ and X only in their vectorized forms. However, the idea of arranging the partial derivatives into a matrix (rather than a vector) is sometimes useful, so we will retain the expression $\frac{\partial F(x)}{\partial x}$ for a function $F : \mathbb{R} \to \mathbb{R}^{m\times p}$. It is worthwhile noticing that $\mathcal{D}F(x)$ and $\frac{\partial F(x)}{\partial x}$

contain the same partial derivatives, but in $\frac{\partial F(x)}{\partial x}$ they are arranged in a $m \times p$ matrix, whereas $\mathcal{D}F(x) = \operatorname{vec} \frac{\partial F(x)}{\partial x}$.

The following theorem transforms the problem of finding the Jacobian matrix of a matrix function into the problem of finding its differential (denoted by d), which is generally easier.

Theorem 1 (Identification theorem). *The following two equations are equivalent:*

$$\mathrm{d} \operatorname{vec} F(X) = A(X) \,\mathrm{d} \operatorname{vec} X \tag{A.2}$$

$$\mathcal{D}F(X) = A(X). \tag{A.3}$$

The vec operator has some interesting properties in connection with the *Kronecker product* $\otimes$, in particular:

$$\operatorname{vec} AXB = (B^\top \otimes A) \operatorname{vec} X. \tag{A.4}$$

This formula and the identification theorem imply that:

$$\mathcal{D}(AXB) = (B^\top \otimes A) \tag{A.5}$$

Theorem 2 (Chain rule). *Let $F : \mathbb{R}^{n\times q} \to \mathbb{R}^{m\times p}$ and $G : \mathbb{R}^{m\times p} \to \mathbb{R}^{r\times s}$ be differentiable functions. If the composite function $H(X) = G(F(X))$ is differentiable at X_0, its Jacobian matrix is*

$$\mathcal{D}H(X_0) = (\mathcal{D}G(Y_0))(\mathcal{D}F(X_0)) \tag{A.6}$$

where $Y_0 = F(X_0)$.

Appendix B. Derivatives of rotation matrices

Let $\mathbf{u} = [u_1, u_2, u_3]^\top$ be a unit vector, and ϑ be an angle. The matrix representing the rotation by ϑ around the axis $\mathbf{u}$ is given by the Rodriguez formula:

$$R(\vartheta, \mathbf{u}) = (I + \sin\vartheta[\mathbf{u}]_\times + (1 - \cos\vartheta)[\mathbf{u}]_\times^2) \tag{B.1}$$

where

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \tag{B.2}$$

Let us now consider the customary representation of rotations with the three Euler angles ω, φ, κ:

$$R(\omega, \varphi, \kappa) = R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2)R(\omega, \mathbf{e}_1) \tag{B.3}$$

where $\mathbf{e}_i$ is the i-th element of the canonical base.

We will now determine the derivative of a rotation matrix wrt the Euler angles:

$$\mathcal{D}R = \mathcal{D}R(\omega, \varphi, \kappa) = \left[\operatorname{vec}\frac{\partial R}{\partial \omega}, \operatorname{vec}\frac{\partial R}{\partial \varphi}, \operatorname{vec}\frac{\partial R}{\partial \kappa}\right]. \tag{B.4}$$

Let us first establish the derivative of a rotation around a given axis. From the Rodriguez formula one gets:

$$\begin{aligned}\frac{\partial R(\vartheta, \hat{\mathbf{u}})}{\partial \vartheta} &= \frac{\mathrm{d}\sin\vartheta}{\mathrm{d}\vartheta}[\mathbf{u}]_\times - \frac{\mathrm{d}\cos\vartheta}{\mathrm{d}\vartheta}[\mathbf{u}]_\times^2 \\ &= \cos\vartheta[\mathbf{u}]_\times + \sin\vartheta[\mathbf{u}]_\times^2\end{aligned} \tag{B.5}$$

Let us focus, e.g., on the first the angle ω that represents a rotation around $\mathbf{e}_1 = [1, 0, 0]^\top$:

$$\begin{aligned}\frac{\partial R(\omega, \varphi, \kappa)}{\partial \omega} &= R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2)\frac{\partial R(\omega, \mathbf{e}_1)}{\partial \omega} \\ &= R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2)(\cos\omega[\mathbf{e}_1]_\times + \sin\omega[\mathbf{e}_1]_\times^2)\end{aligned} \tag{B.6}$$

The derivation for $\dfrac{\partial R(\omega, \varphi, \kappa)}{\partial \varphi}$ and $\dfrac{\partial R(\omega, \varphi, \kappa)}{\partial \kappa}$ is similar. The reader might want to compare these derivative to those reported by (Lucas, 1963), which were based on the observation that (e.g.) $\frac{\partial R(\omega, \mathbf{e}_1)}{\partial \omega} = R(\omega, \mathbf{e}_1)[\mathbf{e}_1]_\times$.