

# BUNDLE BLOCK ADJUSTMENT WITH CONSTRAINED RELATIVE ORIENTATIONS

E. Maset<sup>1</sup>, L. Magri<sup>1</sup>, I. Toschi<sup>2,3</sup>, and A. Fusiello<sup>1\*</sup>

<sup>1</sup> DPIA - University of Udine, Italy - (eleonora.maset, andrea.fusiello)@uniud.it, magri.luca.l@gmail.com

<sup>2</sup> 3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy - tosch@fbk.eu

<sup>3</sup> nFrames GmbH, Stuttgart, Germany - isabella.toschi@nframes.com

## Commission II, WG II/1

**KEY WORDS:** Bundle Adjustment, Orientation, Oblique images, Multi-head camera, Oblique multi-camera

### ABSTRACT:

This paper deals with bundle adjustment with constrained cameras, i.e. where the orientation of certain cameras is expressed relatively to others, and these relative orientations are part of the unknowns. Despite the remarkable interest for oblique multi-camera systems, an empirical study on the effect of enforcing relative orientation constraints in bundle adjustment is still missing. We provide experimental evidence that indeed these constraints improve the accuracy of the results, while reducing the computational load as well. Moreover, we report for the first time in the literature the complete derivation of the Jacobian matrix for bundle adjustment with constrained cameras, to foster other implementations.

## 1. INTRODUCTION

Over the past two decades almost all existing companies in the geospatial sector have opened up for oblique imaging technology and included multi-head oblique camera units in their portfolios. Either as stand-alone solutions (Remondino, Gerke, 2015) or more recently in combination with a LiDAR unit (Toschi et al., 2019), these oblique systems have expanded the potentialities of the area-wide mapping market towards a more complete and intuitive scene understating concept. Indeed, they offer the advantages of a slanted view geometry, that allows the potential 3D reconstruction of building facades and other urban vertical objects (Haala, Rothermel, 2015). However, this comes at the cost of tackling new challenges, that mainly consist in dealing with large variations in image scale and illumination, multiple occlusions and an increased disparity search space. With regards to image block orientation, the main issue here is how to efficiently and rigorously cope with a larger number of unknowns, while minimizing the risk of divergence for the adjustment of the bundle of image rays.

Although the use of a constrained bundle block adjustment (BBA) in oblique acquisition scenarios represents an hot research topic, few of the software applications available to the public allows to enforce relative orientation constraints. As a result, most of the literature (Wiedemann, Moré, 2012, Rupnik et al., 2013, Rupnik et al., 2014) suggests that relative constraints should be enforced without being able to produce experimental evidence of that. (Cavegn et al., 2018) implement relative constraints in a terrestrial mobile mapping application, but do not assess this feature in isolation. They also report that Pix4D is able to enforce relative constraints, but no study in the literature evaluated it so far.

Constrained relative orientations are usually recovered as part of the calibration process of multi-camera rigs (Esquivel et al., 2007, Dai et al., 2009, Schneider, Förstner, 2013). The exterior orientation of the multi-camera system is then treated as a *Non-Perspective-n-Point* problem, where the camera rig is modeled

as a generalized camera (i.e., a camera in which the bundle of rays do not intersect in a single point), and solved exploiting the relative orientations estimated by calibration.

The only work, to the best of our knowledge, that analyses an implementation of BBA with relative orientation constraints and compares the results achieved with the custom BBA is (Sun et al., 2016), that however states that this approach has a lower accuracy on the basis of experiments that report worse reprojection errors than in the case where the constraints are ignored. It must be said, though, that looking at the reprojection error without considering the degrees of freedom of the model being fitted may lead to biased conclusions.

In this paper we will argue and provide evidence that exploiting the rigidity of the system has two benefits: it reduces the number of unknowns *and* it improves the accuracy. As a matter of fact, disregarding a physical constraint that actually holds, artificially increases the degrees of freedom of the system, leading to over-fitting (i.e., fitting the noise).

A further contribution of this paper is to work out in full details the formula for the Jacobian matrix of BBA with constrained cameras, which was missing in the literature so far. We will use the “matrix differential calculus” formalism (Magnus, Neudecker, 1999), which allows a compact and modular derivation.

## 2. PROBLEM STATEMENT

Let us consider an oblique multi-camera system 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, while the remaining two oblique cameras point in opposite directions along-track. The five cameras are housed in a suitable frame that ensures rigidity and

\* Corresponding author

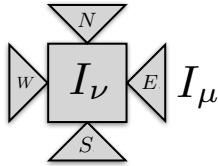


Figure 1. Illustration of the Maltese cross example. In this case  $\tau = E$  and so  $G_\mu = G_E G_\nu$ .

their shutters are synchronized to operate simultaneously<sup>1</sup>.

- Let  $I_\nu$  be one nadir image, whose exterior orientation is  $G_\nu = [R_\nu | \mathbf{t}_\nu]$  ( $G_\nu$  transforms 3D point coordinates from an external (object) reference frame to the camera reference, whose origin is at the centre of projection);
- let  $I_\mu$  be one of the (four) oblique images relative to  $I_\nu$ , whose exterior orientation is  $G_\mu = [R_\mu | \mathbf{t}_\mu]$ ;
- let  $G_\tau = [R_\tau | \mathbf{t}_\tau]$  be the orientation of the oblique image relative to the nadir image ( $G_\tau$  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:  $\tau = \{N, S, E, W\}$  which are fixed but unknown;
- the exterior orientation of the oblique image  $G_\mu$  can be written as a function of those of the nadir image and the relative orientation:

$$G_\mu = G_\tau \cdot G_\nu = [R_\tau \cdot R_\nu | R_\tau \cdot \mathbf{t}_\nu + \mathbf{t}_\tau]; \quad (1)$$

- let us parametrize the nadiral orientation  $G_\nu$  with the 6 parameters  $\mathbf{g}_\nu^\top = [\omega, \varphi, \kappa, \mathbf{t}_\nu]$ , where the first three are the Euler angles that represent the rotation (see App. B);
- let us parametrize the relative orientation  $G_\tau$  with the 6 parameters  $\mathbf{g}_\tau^\top = [\alpha, \beta, \gamma, \mathbf{t}_\tau]$  where the first three are the Euler angles;
- let us write the collinearity equations in compact form as:

$$\mathbf{x} = f(\mathbf{X}, g) = f_P(R(\omega, \varphi, \kappa)\mathbf{X} + \mathbf{t}) \quad (2)$$

where  $\mathbf{X}$  are the coordinates of an object point,  $\mathbf{x}$  are the two coordinates of the corresponding image point,  $g = [\omega, \varphi, \kappa, \mathbf{t}]^\top$  represent the exterior orientation of the image, and the central projection is accomplished by:

$$f_P([x, y, z]^\top) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top. \quad (3)$$

- Interior orientation is assumed known and fixed (however, the estimate of the interior parameters, possibly including distortion parameters, could be added in the process).

### 3. JACOBIAN MATRIX OF CRO-BBA

Performing BBA using any variant of the Gauss-Newton method entails computing the Jacobian matrix that collects the partial derivatives of  $f$ .

<sup>1</sup> Small lags can be absorbed into the relative orientation as translations along the direction of motion

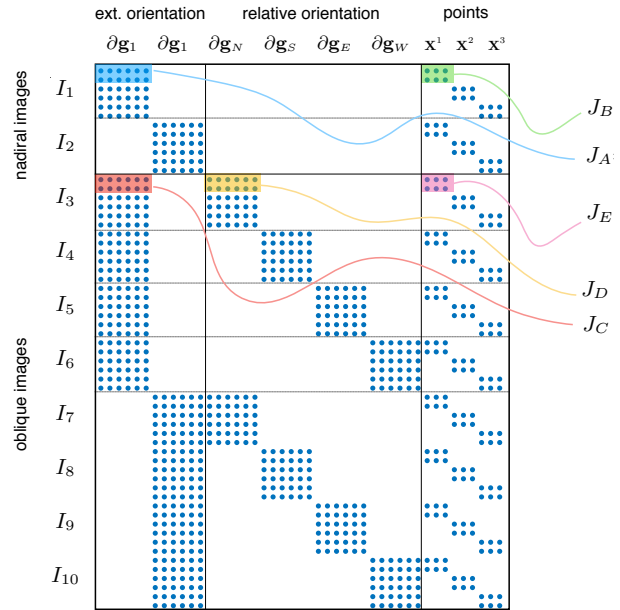


Figure 2. Primary structure of the CRO-BBA Jacobian for a block of 2 nadir cameras, 8 oblique and 3 tie-points.

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 that we are examining in this paper, which will be referred to in the following as ‘‘Constrained Relative Orientations BBA’’ or CRO-BBA for short, 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.

We assume that the oblique images relative to the same nadir image are consecutive and ordered consistently.

Instead of reverting to subscript notation for computing such derivatives, we perform the entire operation using the *matrix differential calculus* introduced by (Magnus, Neudecker, 1999), which we will briefly review in App. A. After some rewriting (reported in App. C) one ends up with the following formulae for the non-zero blocks of the CRO-BBA Jacobian:

$$J_A = \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial \mathbf{g}_\nu^\top} = \mathcal{D}f_P(R_\nu \mathbf{X} + \mathbf{t}_\nu) [(\mathbf{X}^\top \otimes I_3) \mathcal{D}R_\nu | I_3] \quad (4)$$

$$J_B = \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial \mathbf{X}^\top} = \mathcal{D}f_P(R_\nu \mathbf{X} + \mathbf{t}_\nu) R_\nu \quad (5)$$

$$J_C = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\nu^\top} = \mathcal{D}f_P(\mathbf{X}') [(\mathbf{X}^\top \otimes R_\tau) \mathcal{D}R_\nu | R_\tau] \quad (6)$$

$$J_D = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\tau^\top} = \mathcal{D}f_P(\mathbf{X}') [((R_\nu \mathbf{X} + \mathbf{t}_\nu)^\top \otimes I_3) \mathcal{D}R_\tau | I_3] \quad (7)$$

$$J_E = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{X}^\top} = \mathcal{D}f_p(\mathbf{X}') R_\tau R_\nu \quad (8)$$

where  $\mathbf{X}' = R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau$ . The blocks  $J_A$ ,  $J_C$  and  $J_D$  have dimension  $2 \times 6$ , whereas  $J_B$  and  $J_E$  have dimension  $2 \times 3$ .

Using these formulae, we implemented CRO-BBA in MATLAB adopting camera reduction (Brown, 1958) and the Levenberg-Marquardt optimization strategy for free-network adjustment. 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, 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). Classical BBA is obtained by switching off relative orientation constraints in our MATLAB implementation.

#### 4. EXPERIMENTS

Let  $k$  be the number of cameras that compose the multi-head system, let  $m$  be the number of nadir images (the total number of images is  $km$ ), and  $n$  be the number of tie-points. The BBA residual is the  $2kmn \times 1$  vector  $\mathbf{v}$  whose entries are given by  $\mathbf{x}_{ij} - f(\mathbf{X}_i, g_j)$  for  $i = 1 \dots n$ ,  $j = 1 \dots km$ . While the RMS (root mean square error) is appropriate when comparing the results of regression to models with the same degrees of freedom, in our case one should instead look at the root of the reference variance - RRV (Kraus, 2007), computed as:

$$RRV = \sqrt{\frac{\mathbf{v}^\top \mathbf{v}}{\ell - p}} \quad (9)$$

where  $\ell$  is the number of equations (and the length of  $\mathbf{v}$ ) and  $p$  is the number of unknowns.

If all tie-points were seen in all images  $\ell = 2kmn$ , but in practice it is much smaller than that, as it depends on the visibility. As for  $p$ , if the oblique images are adjusted as if their orientations were independent, the system has  $p = 6km + 3n$  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 and memory footprint.

We performed tests on simulated and real data. Simulation used 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 ( $10,328 \times 7,760$  pixels, 50 mm focal length) in a "Maltese cross" arrangement. Angles between the optical axis of the nadir camera and those of the oblique images were set to  $30^\circ$ , 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, with an average of 30 points visible in each image.

Real data were collected by the Leica CityMapper hybrid sensor over the city of Heilbronn, Germany (data courtesy by Leica

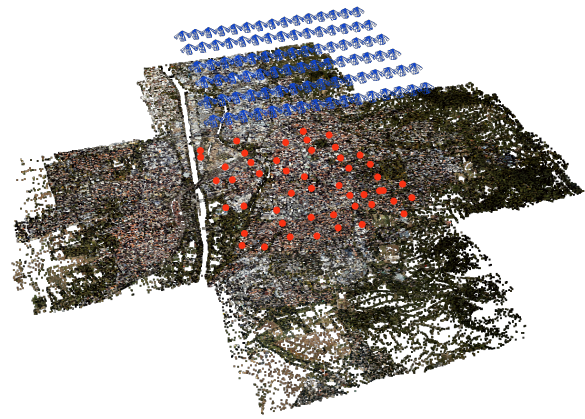


Figure 3. Real data used for the experiment. Cameras are represented by blue pyramids, control points are displayed as red dots (best viewed in color).

Geosystems). Leica CityMapper 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  $10,320 \times 7,752$  pixels, NIR CCD size  $3,336 \times 4,500$  pixels, 83 mm focal length) and four  $45^\circ$ -tilted camera heads (RGB CCD size  $10,320 \times 7,752$  pixels, 156 mm focal length). The flight plan was designed using an average nadir GSD (ground sample distance) 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.  $3.5 \text{ km} \times 3.5 \text{ km}$  over the city centre (Fig. 3). As ground truth data we employed 49 control points, surveyed with RTK GNSS with a mean 3D accuracy of 5 cm.

Besides image-space errors, the evaluation considered the RMS of the residual distances between corresponding 3D points after least-squares (Procrustean) alignment of i) object points to control points; ii) COPs (Centre Of Projection) to their known positions. Please note that all tie-points in the simulated experiments are also control points and that COPs are not known in real experiments<sup>2</sup>.

Since our BBA/CRO-BBA produce free-network solutions and the alignment transformation is a similitude, any non-Euclidean deformation of the model generated by BBA/CRO-BBA is revealed (in object-space) by the alignment residuals.

##### 4.1 Simulated data

To carry out the simulation, we defined the ground truth exterior orientations on the basis of the flight plan described in the previous section. 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  $\sigma = 0.2^\circ$  for the Euler angles. Moreover, we added a random noise with  $\sigma = 0.05$  m and  $\sigma = 0.05^\circ$  to the relative positions and angles of the oblique cameras, respectively. For each setting the trial was run 100 times.

<sup>2</sup> Actually they are known from the onboard GNSS/INS, but their accuracy is not sufficient to qualify them as reference values

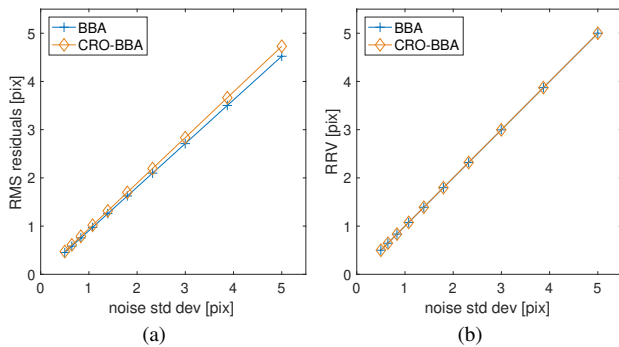


Figure 4. Image-space errors: RMS of the residuals and Root of Reference Variance (RRV) for BBA and CRO-BBA vs image noise. RRV is equal to the standard deviation of the image noise.

Figure 4(a) shows that the RMS of the reprojection errors (averaged on the 100 trials) is slightly larger for CRO-BBA, while the RRV 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. Concluding that CRO-BBA is worse than unconstrained BBA on the basis of the reprojection errors (Sun et al., 2016), however, is wrong, because the latter has more degrees of freedom (d.o.f.) than the former, being thus more capable of reducing the residuals by over-fitting (i.e., fitting the noise). Indeed, when the d.o.f. of the model are taken into account with RRV (Fig. 4(b)), the two are perfectly equivalent in terms of image-space errors.

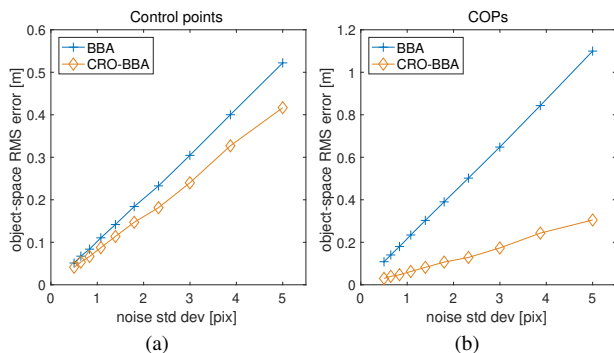


Figure 5. Object-space errors: RMS of the alignment residuals computed on control points (a) and COPs (b) vs image noise.

The advantages of the CRO-BBA can be appreciated from the object-space errors, i.e. the alignment residuals wrt control points (Figs 5 and 6). 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 error can be shifted from COPs to object points by dealing differently with the datum defect, but the essence of this result does not change.

For CRO-BBA, errors affecting control points and COPs are comparable (please note the different scales of y-axes for Fig. 5a and 5b). BBA on the other hand does not impose constraints among the COPs of the same multi-camera, that are free to move: this results in an increased error of the COPs.

In a practical scenario, these residual non-Euclidean deformations would have been compensated for by ground control points, and this is why this effect is seldom detected: using a model

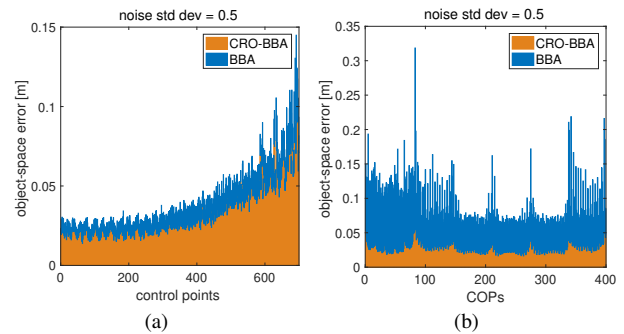


Figure 6. Object-space errors for control points (a) and COPs (b) for 0.5 pixels noise standard deviation. In (a) points are sorted by decreasing ray-multiplicity. This figure is best viewed in colour.

with a number of d.o.f. in excess is balanced by additional constraints.

In Figure 6(b) one might notice that with BBA the first 80 COPs have larger errors. As a matter of fact, the first 80 indices correspond to nadir images, and the average error over them is 0.123 m, while the average error on oblique images is 0.084 m. The two distributions are different with a confidence very close to 100% according to the t-test. So, ignoring relative orientation constraints affects the localization of the nadir images more than the one of oblique images.

Please note that in Fig. 6(a) the points are sorted by decreasing ray-multiplicity, and the errors follow consistently: the lowest error is achieved by the points with highest ray-multiplicity.

## 4.2 Real data

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. For the interior 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 1.02 pixel for BBA and 1.04 pixel CRO-BBA, while the object-space errors wrt control points are shown in Fig. 7(a), where it can be appreciated that the CRO-BBA produces more accurate results than BBA. The RMS of the object-space errors is 0.20 m for BBA and 0.14 m for CRO-BBA, and the t-test confirm that the two distributions are indeed different with 99.85%

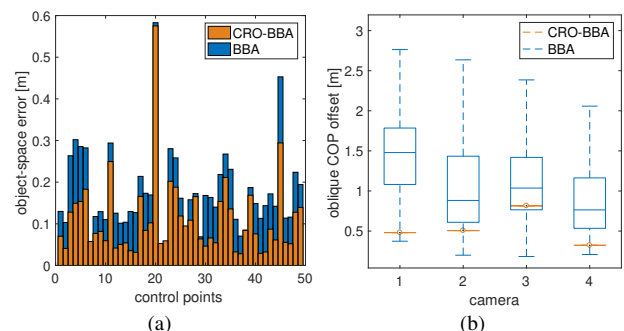


Figure 7. Results of the experiment with real data. (a) Object-space errors for control points; (b) Box plot of the oblique COP offsets. This figure is best viewed in colour.

confidence. This experiment is in agreement with the previous simulations: enforcing relative constraints is beneficial in terms of errors in object-space, notwithstanding the image-space error (RRV) is approximately the same.

Figure 7(b) 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 as an indicator of the precision of the COP localization achieved by BBA. Please note that the scale of y-axis is in meters.

## 5. CONCLUSIONS

We investigated bundle adjustment with constrained cameras, i.e., where the orientation of certain cameras is expressed relatively to others, and these relative orientations are part of the unknowns.

Despite some authors suggest that enforcing relative orientation constraints degrades the accuracy of the results, we argued that indeed this is not the case. On the contrary, exploiting the rigidity of the system reduces the number of unknowns and improves the accuracy, when correctly measured. Experiments with simulated and real data confirm that image-space errors is not significant to discriminate between the two solutions (with and without relative orientation constraints), while the object-space errors demonstrate that the unconstrained solution is affected by severe deformations that impact on the accuracy of the result.

The bottom line is that there is no reason to drop relative orientation constraints in the BBA of image blocks with (synchronized) oblique multi-camera system.

## ACKNOWLEDGMENTS

Luca Magri has been supported by "HEaD Higher Education and Development Project" FP1619942003 and 3Dflow s.r.l. Thanks to Helica s.r.l. for providing the data for the simulations, and to Leica Geosystems for providing the CityMapper dataset.

## 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(144), 396-415.

Cavegn, S., Blaser, S., Nebiker, S., Haala, N., 2018. Robust and accurate image-based georeferencing exploiting relative orientation constraints. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2), 57-64.

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

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

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

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(4), 708-715.

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

Remondino, F., Gerke, M., 2015. Oblique aerial imagery – a review. *Proc. Photogrammetric Week 2015*, 75-83.

Rupnik, E., Nex, F., Remondino, F., 2013. Automatic orientation of large blocks of oblique images. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(Part 1/W1), 299-304.

Rupnik, E., Nex, F., Remondino, F., 2014. Oblique multi-camera systems-orientation and dense matching issues. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3(Part 1), 107-114.

Schneider, J., Förstner, W., 2013. Bundle Adjustment and System Calibration with Points at Infinity for Omnidirectional Camera Systems. *Photogrammetrie-Fernerkundung-Geoinformation*, 2013(4), 309-321.

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. *Proceedings of the International Workshop on Vision Algorithms*, Springer-Verlag, 298-372.

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

## A. MATRIX DIFFERENTIAL CALCULUS

While for a differentiable function  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  the partial derivative are customary packed into a matrix called Jacobian, for matrix functions  $\mathbb{R}^{n \times q} \rightarrow \mathbb{R}^{m \times p}$  the question naturally arises how to order the  $mnpq$  partial derivatives, for this can be done in many ways. (Magnus, Neudecker, 1999) argue that the following notation should be used, for a number of good reasons, the most important being that it allows to use 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

$$DF(X) = \frac{\partial \text{vec } F(X)}{\partial (\text{vec } X)^T}. \quad (10)$$



where  $\text{vec}(A)$  is the column vector obtained by stacking the columns of  $A$ . Note that (10) also defines the Jacobian matrix for vector functions of vector variables.

Definition 4 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  $F: \mathbb{R} \rightarrow \mathbb{R}^{m \times p}$ . 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  $\mathcal{D}F(x) = \text{vec} \frac{\partial F(x)}{\partial x}$ .

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:*

$$d \text{vec} F(X) = A(X) d \text{vec} X \quad (11)$$

$$\mathcal{D}F(X) = A(X). \quad (12)$$

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

$$\text{vec} AXB = (B^\top \otimes A) \text{vec} X. \quad (13)$$

This formula and the identification theorem imply that:

$$\mathcal{D}(AXB) = (B^\top \otimes A) \quad (14)$$

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

$$\mathcal{D}H(X_0) = (\mathcal{D}G(Y_0))(\mathcal{D}F(X_0)) \quad (15)$$

where  $Y_0 = F(X_0)$ .

## B. DERIVATIVES OF THE EULER ROTATION MATRIX

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

$$R(\vartheta, \mathbf{u}) = (I + \sin \vartheta [\mathbf{u}]_\times + (1 - \cos \vartheta) [\mathbf{u}]_\times^2) \quad (16)$$

where

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \quad (17)$$

Let us now consider the customary representation of rotations with the three Euler angles  $\omega, \varphi, \kappa$ :

$$R(\omega, \varphi, \kappa) = R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2)R(\omega, \mathbf{e}_1) \quad (18)$$

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:

$$\mathcal{D}R = \mathcal{D}R(\omega, \varphi, \kappa) = \left[ \text{vec} \frac{\partial R}{\partial \omega}, \text{vec} \frac{\partial R}{\partial \varphi}, \text{vec} \frac{\partial R}{\partial \kappa} \right]. \quad (19)$$

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

$$\begin{aligned} \frac{\partial R(\vartheta, \hat{\mathbf{u}})}{\partial \vartheta} &= \frac{d \sin \vartheta}{d \vartheta} [\mathbf{u}]_\times - \frac{d \cos \vartheta}{d \vartheta} [\mathbf{u}]_\times^2 \\ &= \cos \vartheta [\mathbf{u}]_\times + \sin \vartheta [\mathbf{u}]_\times^2 \end{aligned} \quad (20)$$

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

$$\begin{aligned} \frac{\partial R(\omega, \varphi, \kappa)}{\partial \omega} &= R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2) \frac{\partial R(\omega, \mathbf{e}_1)}{\partial \omega} \\ &= R(\kappa, \mathbf{e}_3)R(\varphi, \mathbf{e}_2) (\cos \omega [\mathbf{e}_1]_\times + \sin \omega [\mathbf{e}_1]_\times^2) \end{aligned} \quad (21)$$

The derivation for  $\frac{\partial R(\omega, \varphi, \kappa)}{\partial \varphi}$  and  $\frac{\partial R(\omega, \varphi, \kappa)}{\partial \kappa}$  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.)  $\frac{\partial R(\omega, \mathbf{e}_1)}{\partial \omega} = R(\omega, \mathbf{e}_1) [\mathbf{e}_1]_\times$ .

## C. DERIVATIVES OF THE COLLINEARITY EQUATIONS

Before working out the derivative of the function  $f$  defined in (2), let us establish the Jacobian of the projection function  $f_p$  by differentiating it element-by-element:

$$\mathcal{D}f_p(\mathbf{W}) = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix}, \quad \mathbf{W}^\top = [x, y, z]; \quad (22)$$

We now proceed to derive the expression for the non-zero blocks of the BBA Jacobian, using the formalism outlined in App. A. Blocks  $J_A$  and  $J_B$  correspond to nadir images, and they are easily derived:

$$\begin{aligned} J_A &= \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial \mathbf{g}_\nu^\top} = \frac{\partial f_p(\mathbf{W})}{\partial \mathbf{W}^\top} \Big|_{\mathbf{W}=R_\nu \mathbf{X} + t_\nu} \frac{\partial (R_\nu \mathbf{X} + t_\nu)}{\partial \mathbf{g}_\nu^\top} \\ &= \mathcal{D}f_p(R_\nu \mathbf{X} + t_\nu) [(\mathbf{X}^\top \otimes I_3) \mathcal{D}R_\nu | I_3] \end{aligned} \quad (23)$$

$$\begin{aligned} J_B &= \frac{\partial f(\mathbf{X}, \mathbf{g}_\nu)}{\partial \mathbf{X}^\top} = \frac{\partial f_p(\mathbf{W})}{\partial \mathbf{W}^\top} \Big|_{\mathbf{W}=R_\nu \mathbf{X} + t_\nu} \frac{\partial (R_\nu \mathbf{X} + t_\nu)}{\partial \mathbf{X}^\top} \\ &= \mathcal{D}f_p(R_\nu \mathbf{X} + t_\nu) R_\nu \end{aligned} \quad (24)$$

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

$$J_D = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\mu^\top} = \left[ \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial [\alpha, \beta, \gamma]} \Big| \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\mu^\top} \right] \quad (25)$$

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

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\alpha, \beta, \gamma]} &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau)}{\partial[\alpha, \beta, \gamma]} \\ &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau (R_\nu \mathbf{X} + \mathbf{t}_\nu))}{\partial[\alpha, \beta, \gamma]} \quad (26) \\ &= \mathcal{D}f_p(\mathbf{X}') ((R_\nu \mathbf{X} + \mathbf{t}_\nu)^\top \otimes I_3) \mathcal{D}R_\tau \end{aligned}$$

and also

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\tau^\top} &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau)}{\partial \mathbf{t}_\tau^\top} \\ &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial \mathbf{t}_\tau}{\partial \mathbf{t}_\tau^\top} \quad (27) \\ &= \mathcal{D}f_p(\mathbf{X}') I_3 \end{aligned}$$

Now we derive with respect to the nadiral orientation:

$$J_C = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{g}_\nu^\top} = \left[ \begin{array}{c|c} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\omega, \varphi, \kappa]} & \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\nu^\top} \end{array} \right] \quad (28)$$

where

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial[\omega, \varphi, \kappa]} &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau)}{\partial[\omega, \varphi, \kappa]} \\ &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial R_\tau R_\nu \mathbf{X}}{\partial[\omega, \varphi, \kappa]} \quad (29) \\ &= \mathcal{D}f_p(\mathbf{X}') (\mathbf{X}^\top \otimes R_\tau) \mathcal{D}R_\nu \end{aligned}$$

and

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{t}_\nu^\top} &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau)}{\partial \mathbf{t}_\nu^\top} \\ &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial R_\tau \mathbf{t}_\nu}{\partial \mathbf{t}_\nu^\top} \quad (30) \\ &= \mathcal{D}f_p(\mathbf{X}') R_\tau \end{aligned}$$

Finally, for the 3D point:

$$\begin{aligned} J_E = \frac{\partial f(\mathbf{X}, \mathbf{g}_\mu)}{\partial \mathbf{X}^\top} &= \mathcal{D}f_p(\mathbf{X}') \frac{\partial(R_\tau R_\nu \mathbf{X} + R_\tau \mathbf{t}_\nu + \mathbf{t}_\tau)}{\partial \mathbf{X}^\top} \quad (31) \\ &= \mathcal{D}f_p(\mathbf{X}') R_\tau R_\nu \end{aligned}$$

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. Interior orientation can be easily catered for as well.