EXTERNAL VERIFICATION OF THE BUNDLE ADJUSTMENT IN PHOTOGRAMMETRIC SOFTWARE USING THE DAMPED BUNDLE ADJUSTMENT TOOLBOX

Niclas Börlin[a,*], Pierre Grussenmeyer[b]

[a] Dept. of Computing Science, Umeå University, Sweden, niclas.borlin@cs.umu.se
[b] ICube Laboratory UMR 7357, Photogrammetry and Geomatics Group, INSA Strasbourg, France,
pierre.grussenmeyer@insa-strasbourg.fr

Commission V, WG V/1

KEY WORDS: Software, Bundle adjustment, Accuracy, Quality, Best practice, Distortion, Data sets

ABSTRACT:

The aim of this paper is to investigate whether the Matlab-based Damped Bundle Adjustment Toolbox (DBAT) can be used to provide independent verification of the BA computation of two popular software — PhotoModeler (PM) and PhotoScan (PS).
For frame camera data sets with lens distortion, DBAT is able to reprocess and replicate subsets of PM results with high accuracy. For lens-distortion-free data sets, DBAT can furthermore provide comparative results between PM and PS. Data sets for the discussed projects are available from the authors.
The use of an external verification tool such as DBAT will enable users to get an independent verification of the computations of their software. In addition, DBAT can provide computation of quality parameters such as estimated standard deviations, correlation between parameters, etc., something that should be part of best practice for any photogrammetric software. Finally, as the code is free and open-source, users can add computations of their own.

1. INTRODUCTION

Contemporary commercial photogrammetric software have many advantages such as streamlined work-flows, automatic point detection, 3D model visualization, etc. However, a drawback of most commercial software is the lack of transparency of what computations are taking place or even what mathematical problem is being solved.

At the core of most photogrammetric software is the Bundle Adjustment (BA) procedure. Most software cite the collinearity equations/pin-hole camera model (see e.g. Mikhail et al., 2001, Ch. 4.5.1; Hartley and Zisserman, 2003, Ch. 6) and the Brown lens distortion model (Brown, 1971) as their mathematical foundation. Given the well-known theory, any BA implementation should produce the same results given the same input. However, as many users of photogrammetric software can testify, this is not necessarily the case. Or at least, it is hard to verify why two different pieces of software generate slightly (or very) different results. This may be especially true if the software in question are designed from the differing frames of mind of Photogrammetry and Computer Vision.

The aim of this paper is to investigate whether the Matlab-based Damped Bundle Adjustment Toolbox (DBAT) (Börlin and Grussenmeyer, 2013, 2014) can be used to provide independent verification of the BA computation of two popular software — PhotoModeler (PM) and PhotoScan (PS).

2. THEORY

This section describes the underlying theory, including some varying interpretations and other sources of possible confusion.

2.1 Collinearity

The collinearity equations are typically described in photogrammetric textbooks (e.g. Kraus, 1993, eq. (5.3-4); Mikhail et al., 2001, eq. (4-24); McGlone et al., 2004, eq. (3.132–3.133); Luhmann et al., 2006, eq. (4.8))

$$x - x_0 = -f\frac{r_{11}(X-X_0)+r_{12}(Y-Y_0)+r_{13}(Z-Z_0)}{r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)}, \tag{1a}$$

$$y - y_0 = -f\frac{r_{21}(X-X_0)+r_{22}(Y-Y_0)+r_{23}(Z-Z_0)}{r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)}, \tag{1b}$$

where the object point (OP) with coordinates (X, Y, Z) is projected through the center of projection (X_0, Y_0, Z_0) to the image coordinates (x, y) in a camera with focal length f (sometimes called camera constant or principal distance) and principal point at image coordinates (x_0, y_0). The rotation matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{2}$$

describe the rotation from the object space coordinate system to the camera coordinate system. Variants of eq. (1) include whether the principal point is explicitly included in the equations and the sign of the internal parameters.

In the Computer Vision literature, the collinearity conditions are sometimes referred to as the *pinhole camera model*. Using homogeneous coordinates, a projection of the 3D point $\mathbf{X}$ through a camera center at $C = (X_0, Y_0, Z_0)^T$ is written as (cf. e.g. eq. (6.11) in Hartley and Zisserman (2003) and eq. (3.131) in McGlone et al. (2004))

$$\mathbf{x} = KR\begin{pmatrix} I_3 & -C \end{pmatrix}\mathbf{X}, \tag{3}$$

[*] Corresponding author

where the upper-triangular camera calibration matrix K contain the camera-internal (intrinsic) parameters and the matrix R is the same as in eq. (2).

2.2 The rotation

A rotation in three dimensions has three degrees of freedom. Common parameterizations of the rotation matrix R include some sequence of Euler angles, Rodriguez parameters, or quaternions (Wolf and Dewitt 2000, Ch. 10; Mikhail et al. 2001, App. E; McGlone et al. 2004, Ch. 2.1.2). With few exceptions, the choice of parameterization should not affect the estimated rotation. However, it is non-trivial to convert the parameters between different parameterization to e.g. compare the values estimated by different software.

2.3 Camera units

If the image measurements are performed in digital images, the most common measurement unit is the pixel. If the internal camera parameters are expressed in physical units, e.g. mm, a scaling is necessary. In Computer Vision, this scaling is commonly factored into the camera calibration matrix K, and the camera-internal parameters are expressed in units of pixels. In Photogrammetry it is customary to retain the connection to the physical focal length of the camera and treat the conversion to pixels are a separate step.

Mathematically, the two methods are identical. However, if a pre-calibrated camera is to be used in multiple software that uses different camera units, a potential error source is introduced.

2.4 Image coordinate system

A related issue is what image coordinate system is used, i.e. where is the origin, what is the axis order (row-column vs. the Cartesian x-y), what direction is positive, etc. Furthermore, for digital images, does the center of a pixel or the border of a pixel lie on integer coordinates? This is especially important to get right if image measurements from two different systems are to be used together.

2.5 Right-handed vs. left-handed coordinate systems

If the image coordinates are measured in the Cartesian x-y system (the origin in the lower left corner, positive x to the right, positive y up), an OP in front of the camera will have negative z coordinates in the 3D camera coordinate system if the right-hand axis convention is followed. While such a mental image might make sense for aerial projects (looking “down” at the Earth), for close-range projects the mental model that a positive depth coordinate corresponds to points in front of the camera might be preferred. If the latter is chosen, a mirroring must be introduced at some part of the computational chain, either by mirroring the object space or using a left-handed image coordinate system with e.g. positive y being defined as down.

2.6 Lens distortion

The Brown lens distortion model (Brown, 1971) separates the effects of lens distortion into a radial and tangential component. The radial component is usually presented as

$$x_c = x_m + (x_m - x_p)(K_1 r^2 + K_2 r^4 + \cdots), \tag{4a}$$

$$y_c = y_m + (y_m - y_p)(K_1 r^2 + K_2 r^4 + \cdots), \tag{4b}$$

where $r = \sqrt{(x_m - x_p)^2 + (y_m - y_p)^2}$ is the radial distance from the principal point (x_p, y_p). However, while the common photogrammetric interpretation (e.g. Wolf and Dewitt, 2000, Ch. 4.13) is that eq. (4) is a *correction* that is applied to measured coordinates (x_m, y_m) to get corrected coordinates (x_c, y_c), some computer vision authors (e.g. Zhang, 2000, near eq. (11)) suggests that

$$x_d = x_i + (x_i - x_p)(K_1 r^2 + K_2 r^4 + \cdots), \tag{5a}$$

$$y_d = y_i + (y_i - y_p)(K_1 r^2 + K_2 r^4 + \cdots), \tag{5b}$$

where $r = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2}$, describe the *distortion* that modifies the ideal coordinates (x_i, y_i) to distorted coordinates (x_d, y_d) during the imaging process. This distinction makes the two interpretations each other's inverses. As this affects the definition of r, any coefficients $K_1, K_2, \ldots$ computed using one interpretation cannot be easily converted to the corresponding coefficients using the other interpretation. A similar reasoning applies to the tangential component.

2.7 Bundle adjustment

Bundle adjustment (BA), sometimes *bundle block adjustment*, is the process of simultaneous estimation of all parameters relevant for a 3D reconstruction from image measurements (Brown, 1976; Mikhail et al., 2001, Ch. 5.8). The types of parameters to be estimated may include the camera internal orientation (IO), the camera exterior orientation (EO, camera position and rotation), the object point (OP) coordinates. The IO parameters include the focal length, principal point, lens distortion parameters and any other parameters that describe the projection inside the camera. The EO parameters include the camera position and the rotation parameters. If the IO parameters are estimated, the process is sometimes called self-calibration (Mikhail et al., 2001, Ch. 5.9; Luhmann et al., 2006, Ch. 4.3.2.4) or auto-calibration (Hartley and Zisserman, 2003, Ch. 19).

The standard photogrammetric BA formulation (McGlone et al., 2004, Ch. 2.2.4) uses the collinearity equations (1) as the functional model and a Gaussian stochastic model, where the m observations in the vector $\mathbf{b}$ typically include image measurement, but may also include IO, EO, or OP observations. The observation covariances are usually written as

$$C_{\mathbf{bb}} = \sigma_0^2 C_{\mathbf{bb}}^{(0)}, \tag{6}$$

where the a priori covariance matrix $C_{\mathbf{bb}}^{(0)}$ is assumed to be known but the standard deviation of unit weight, σ_0, is not. If multiple, mutually independent, observation types are included (image points IP, OP, EO, IO), the a priori covariance matrix takes the form

$$C_{\mathbf{bb}}^{(0)} = \begin{pmatrix} \sigma_{IP}^2 C_{IP}^{(0)} & & & \\ & \sigma_{OP}^2 C_{OP}^{(0)} & & \\ & & \sigma_{EO}^2 C_{EO}^{(0)} & \\ & & & \sigma_{IO}^2 C_{IO}^{(0)} \end{pmatrix}, \tag{7}$$

where the individual blocks are the a priori covariance estimates of each measurement type.

The BA procedure estimates the n unknowns in the vector $\hat{\mathbf{x}}$ by minimizing the quadratic form

$$\Omega^2 = \hat{\mathbf{v}}^T W \hat{\mathbf{v}}, \tag{8}$$

where the residuals $\hat{\mathbf{v}} = \hat{\mathbf{b}} - \mathbf{b}$ are weighted by the weight matrix $W = (C_{\mathbf{bb}}^{(0)})^{-1}$. Furthermore, the standard deviation of unit weight σ_0 is estimated by

$$\widehat{\sigma_0}^2 = \frac{\hat{\mathbf{v}}^T W \hat{\mathbf{v}}}{r}, \tag{9}$$

where the redundancy r is defined as

$$r = m - n. \tag{10}$$

If the assumed covariances in eq. (7) are correct, the estimated $\widehat{\sigma_0}$ will be close to unity. If $\widehat{\sigma_0}^2$ is far from unity, this may be taken as an indication that the individual variance components σ_{IP}^2, σ_{OP}^2, etc., are not properly scaled. In this case, they may be re-estimated from the residuals $\hat{\mathbf{v}}$ (McGlone et al., 2004, Ch. 2.2.4.5) to form

$$C_{\mathbf{bb}}^{(1)} = \begin{pmatrix} \hat{\sigma}_{IP}^2 C_{IP}^{(0)} & & & \\ & \hat{\sigma}_{OP}^2 C_{OP}^{(0)} & & \\ & & \hat{\sigma}_{EO}^2 C_{EO}^{(0)} & \\ & & & \hat{\sigma}_{IO}^2 C_{IO}^{(0)} \end{pmatrix}. \tag{11}$$

The BA process is then repeated with the new weight matrix $W = (C_{\mathbf{bb}}^{(1)})^{-1}$.

Besides the estimated parameters values $\hat{\mathbf{x}}$, a photogrammetric BA algorithm usually computes the covariance of the estimated parameters

$$\widehat{C}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \widehat{\sigma_0}^2 (A^T W A)^{-1}, \tag{12}$$

where A is the Jacobian matrix evaluated for the optimal parameters $\hat{\mathbf{x}}$. The matrix $\widehat{C}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ can be used to compute posterior standard deviations for the estimated parameters and correlations between them.

BA was developed in photogrammetry during the 1950-ies and was made popular within Computer Vision by the paper by Triggs et al. (2000). In contrast to the photogrammetric formulation in eq. (8), Triggs et al. (2000) advocated the use of non-Gaussian distribution models. The choice of error model in the objective function to minimize in a particular software will of course often lead to different "optimal" values.

Depending on author, the BA procedure may include automatic detection and removal of blunders and/or unstable parameters due to high correlations. For this paper, we consider the "core" bundle process without any automatic removal of blunders or parameters.

2.8 Iteration sequence

The BA is a non-linear procedure. Given initial values, the objective function is linearized, and an improved estimate are sought. The process is then repeated "until convergence". Different techniques have been suggested to modify the iteration sequence to improve the convergence properties (see e.g. Lourakis and Argyros, 2009; Börlin and Grussenmeyer, 2013). However, assuming the mathematical problem is well posed, the optimal values returned by a BA procedure should not depend on what particular sequence of computations resulted in those estimates. As with any numerical iterative technique, the exact termination criteria used will affect the exact numerical values of the optimal estimates. The effect should however be small and insignificant.

3. SOFTWARE

3.1 PhotoModeler and PhotoScan

The software that were used in this paper were the EOS Systems PhotoModeler Scanner (PM) v2016.0.5 (Feb 2016)[1] and AgiSoft PhotoScan (PS) v1.2.3 (Jan 2016)[2].

PM is a typical Photogrammetric tool in several respects. It uses the "correction" lens distortion model (4) (EOS Systems, 2016), the internal camera unit is mm, and it uses photogrammetric terminology (*external orientation*, *control points*, etc.). In contrast, PS has an clear Computer Vision heritage: The cited camera model is the "distortion" model (5) (Agisoft, 2016, App. C), the internal camera unit is pixels, and it uses Computer Vision terminology (*extrinsic camera parameters*, *reprojection error*, etc.).

Both software have support for manual and automatic measurements. However, the typical PS project is describes the typical Structure-from-Motion work-flow with mostly an automatic process (Tomasi and Kanade, 1992; Huang and Netravali, 1994). In fact, the camera network cannot be easily performed without automatic point detection and matching. The use of a calibrated camera is described as optional. In contrast, the typical PM project contains many manual steps and starts with a calibrated camera.

Both software allow input of control points (called "markers" in PS) with given isotropic or an-isotropic standard deviations. PM can handle fixed control points, PS cannot. The rotation matrix (2) parameterizations are different: PM uses the ω-ϕ-κ (omega-phi-kappa) angles, PS uses yaw-pitch-roll angles. A minor difference is that PM has integer coordinates at the pixel corners whereas PS has the pixel center at integer coordinates.

Both software can generate report files and export point tables, etc. The biggest difference is that PS does not present any quality parameters such as ray angles, posterior standard deviations nor high correlations.

PM uses a proprietary binary file format. In contrast, PS uses standard files formats: The PS archive files (.psz) files are actually ZIP archives. The archive files contain a main XML document and several binary PLY files with image points and computed object points.

3.2 The Damped Bundle Adjustment Toolbox

The Damped Bundle Adjustment Toolbox (DBAT) (Börlin and Grussenmeyer, 2013, 2014) is a Matlab-based open-source tool for bundle adjustment computations. DBAT can import and process PM projects exported with the "Export Text File" command. In order to compare the bundle results, a Project Status Report and 2D and 3D point tables must furthermore be exported from PM. DBAT can read native PS archive files (.psz).

[1] http://www.photomodeler.com
[2] http://www.agisoft.com

										ray count		ray angle [deg]	
Label	Data set	Software	Images	CP	OP	σ_{CP}	m	n	r	min–max	avg	min–max	avg
C1	CAM	PM	21	4	96	-	4148	414	3734	16–21	20.7	80–90	86
C2	CAM	PM	21	4	96	1 mm	4160	426	3734	16–21	20.7	80–90	86
S1	SXB	PM	5	16	-	-	94	30	64	-	-	-	-
S2	SXB	PM	5	16	-	(2,2,4) cm	142	78	64	-	-	-	-
S3	SXB	PM	5	16	1	(2,2,4) cm	150	81	69	4–4	4.0	26–26	26
S4	SXB	PM	5	16	368	(2,2,4) cm	2440	1173	1267	3–4	3.1	13–26	15
S5	SXB	PS	5	16	1166	(2,2,4) cm	8180	3576	4604	3–5	3.4	13–26	17

Table 1. Experiment details. The number of control points (CP) and object points (OP) are given together with the a priori standard deviation σ_{CP} for the control points. Furthermore, the total number of observations m and unknowns n used to compute the redundancy r is given. Ray counts and ray angle statistics for the OPs are also given.

4. DATA SETS

Two data sets were used in this paper, CAM and SXB.

4.1 CAM

The CAM network consisted of a convergent 21-image close-range data set of the PM camera calibration target (figures 2a, 4ab). The camera was an Olympus C4040Z with 7.5 mm focal length, an image size 2272-by-1704 pixels, and a sensor pixel size of 3.2 microns. Four circular targets were used as CPs. The remaining 96 targets were used as OPs. All targets were measured automatically with the PM sub-pixel target measurement technique.

4.2 SXB

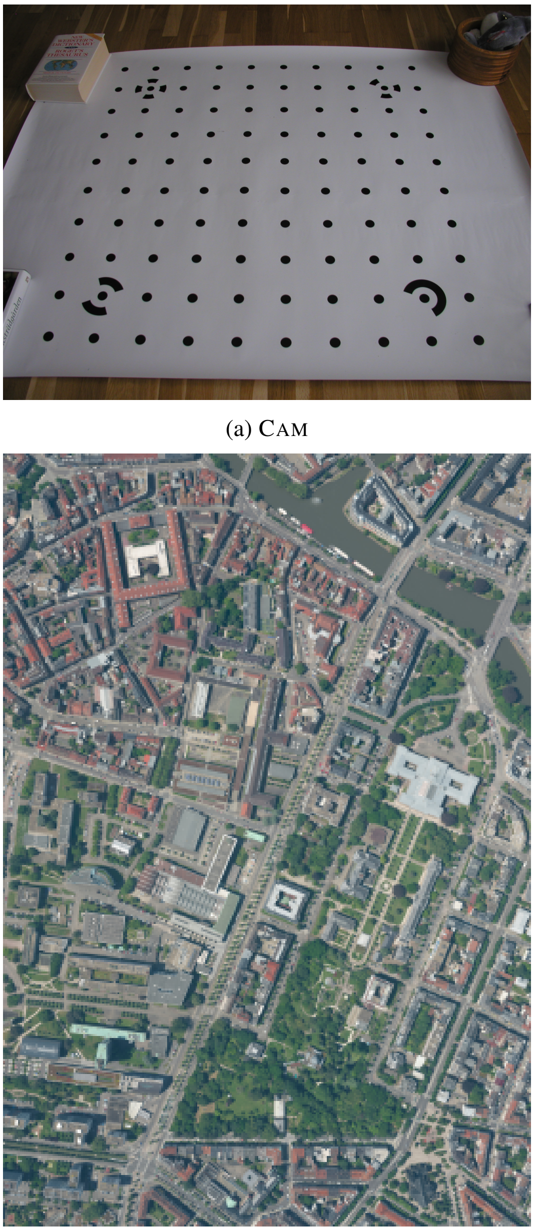
The SXB network consisted of a 5-camera subset of an aerial project of the city of Strasbourg (figures 2b, 4cde). The DiMAC aerial camera was pre-calibrated with a focal length of 123.94 cm and 6 micron pixel size. The images were distortion-free at 8858-by-12996 pixels with approximately 75% overlap and 40% side-lap. The flight height was about 1800 m above ground. A project was created in each of PM and PS. The same 16 control points were measured manually in either software. Each project was extended with automatic points by either software (SmartPoints and Tie points, respectively). After removal of two-ray points, 368 SmartPoints and 1166 Tie points remained in the respective projects. A further object point was manually measured in PM.

5. EXPERIMENTS

Six experiments were set up in PM using both data sets. A seventh experiment was set up in PS using the SXB data set. See Table 1 for details. The five first experiments used the same measurements and were setup to compare the PM and DBAT BA algorithms given the same measurements. The C1 and C2 experiments used all available CP and OP in the CAM data set. In C1, the CP were treated as fixed whereas the CPs were weighted in C2. Similarly, the S1 and S2 used all available CP (no OP) as fixed and weighted, respectively. The S3 experiment was a extension of S2 with a single OP.

The final two experiments, S4 and S5, were set up to compare the PM and PS computational pipeline given the same image data as well as to compare the results with DBAT.

Each PM experiment was run in two versions; (a) one where the bundle adjustment process was preceded by an orientation, and (b) one when it was not.


(a) CAM

(b) SXB

Figure 2. An image from each data set. (a) A close-range image of the PM camera calibration target. (b) An aerial image of the city of Strasbourg.

Each of the PM experiments used the following procedure:

PM.1 Load the project file into PM.

PM.2 For weighted control points: Reset the control point standard deviations to their a priori values (Table 1) (`Marking/Mark/Pin Imports Mode/Convert Import/Set control precision`). Otherwise, the a posteriori values from the previous processing would be used.

PM.3 Select `Project/Process`.

PM.4 Either

(a) Uncheck the `Orientation` checkbox, or

(b) Check the `Orientation` checkbox.

PM.5 `Process` the project.

PM.6 Save the `Project Status Report` to a text file.

PM.7 Export the project using the `Export Text File` command.

PM.8 Open a (3D) `Point Table` and export it to a text file.

PM.9 Open a `2D Point Table` and export it to a text file.

All default settings were used, including the point standard deviation "weights" of 0.1 pixels for sub-pixel targets and 1 pixel for manually measured points and smartpoints.

After the steps above were finished, the following steps were performed by DBAT in Matlab:

DB.1 Load the file exported in step PM.7.

DB.2 Run a spatial resection (Haralick et al., 1994) to establish initial values for the EO parameters based on the given CP coordinates.

DB.3 Run a forward intersection to get initial values for all (non-CP) OP.

DB.4 Run the BA using Armijo line-search (Börlin and Grussenmeyer, 2013).

DB.5 Compute $\widehat{\sigma_0}$ (eq. (9)) and the posterior covariance matrix $\widehat{C}_{\hat{x}\hat{x}}$ (eq. (12)).

DB.6 Compare the estimated CP and OP values and posterior standard deviations with the values in the file from step PM.8.

DB.7 Compare the computed $\widehat{\sigma_0}$, EO values, EO posterior standard deviations and high correlation values (above 95%) with the corresponding values in the `Project Status Report` from step PM.6.

DB.8 Compare the estimated image residuals with the corresponding values in file from step PM.9.

DB.9 Recompute a $\widehat{\sigma_0}$ value based on the residuals loaded from the PM files and equations (9).

DB.10 Record the number of stages that PM used in its processing.

The following procedure was used in PS in the final S5 experiment:

PS.1 Load the project.

PS.2 Select all cameras.

PS.3 Select `Reset Camera Alignment`.

PS.4 With all cameras still selected, select `Align Selected Cameras`.

PS.5 Save the project.

PS.6 Generate a report (`File/Generate Report...`).

The default accuracy values were used: 0.1 pixels for manually measured markers, 1 pixel for automatically measured tie points.

The DBAT procedure for experiment S5 followed steps DB.1–DB.6, except that the data was loaded from the PS project file from step PS.5.

6. RESULTS

The reconstructed camera networks are shown in Figure 4. Detailed difference for $\widehat{\sigma_0}$, EO, OP, CP, and residuals are listed in Table 3. Values differing less than 5×10^{-4} meters, degrees, or pixels, respectively, were considered equal. Furthermore, the differences for "equal" values were all below 0.1 their respective posterior standard deviation σ_{post}. With the exception of experiment S5, the differences classified as "non-equal" were below $0.5\sigma_{\text{post}}$. The S5 differences were above $1.7\sigma_{\text{post}}$.

We observe that experiments S1 with fixed CPs has a difference of about 25% in the estimated $\widehat{\sigma_0}$ despite equal residuals. This difference is reduced to below 0.1% if the redundancy r used in eq. 9 is inflated by 3 times the number of control points. If the same correction is applied to experiment C1, the $\widehat{\sigma_0}$ difference is reduced from 0.2% to below 0.01%.

In addition to the results in Table 3, all high correlations (above 95%) were equal to the number of available digits (0.1 %-units) for all experiments. Furthermore, if the PM/DBAT $\widehat{\sigma_0}$ ratios were factored out, all PM/DBAT estimates of all posterior standard deviations were equal to the number of available digits.

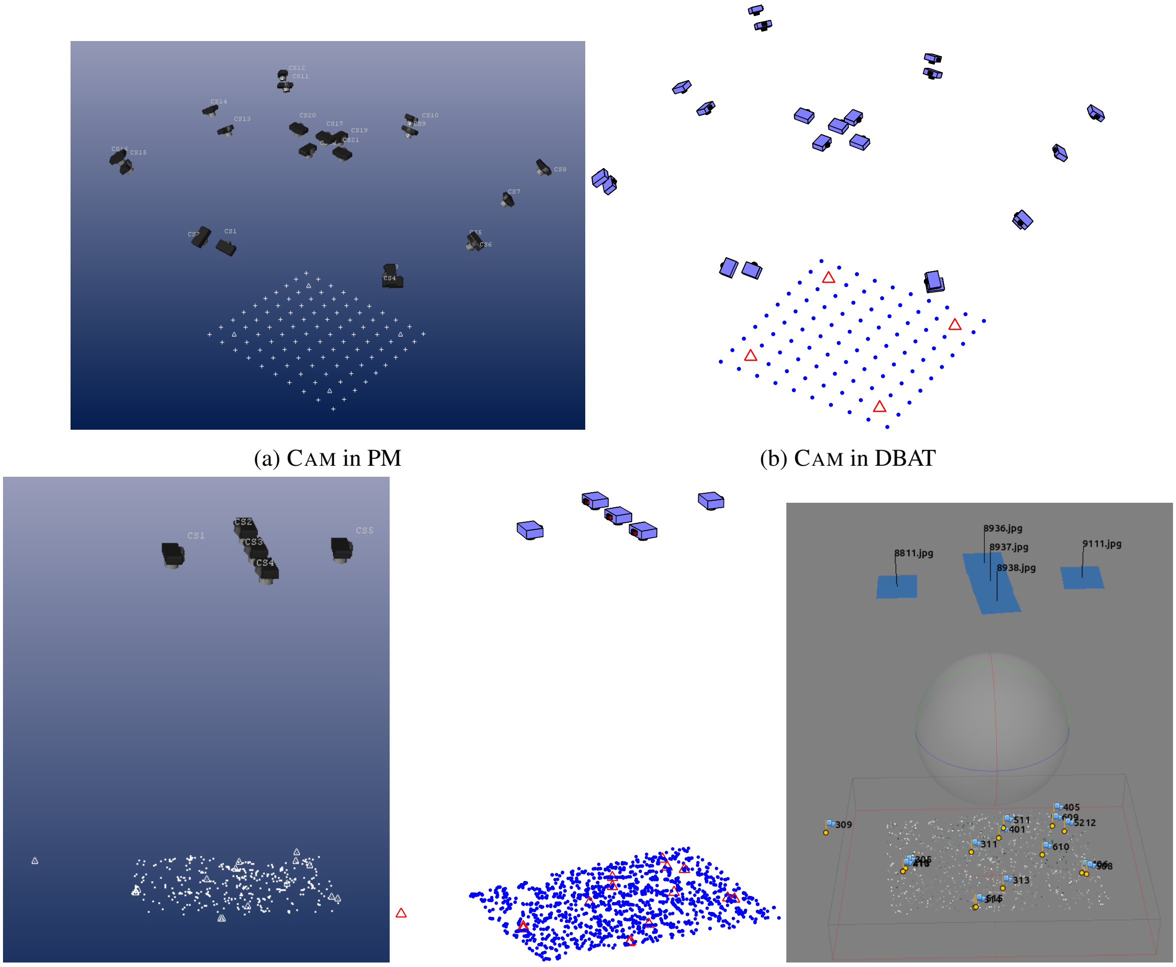
In all experiments where PM performed a one-stage optimization and no orientation (experiments S1B, S2B, S3B), the DBAT values were equal to the PM values, including the image residuals. This was true for both fixed (S1B) and weighted (S2B) control points, and with object points (S3B). The same was true for the S1A experiment with fixed control points. However, for the SXB experiments with weighted control points, whenever PM did a two-stage optimization and/or an orientation procedure was used, the results differed by up to half the posterior standard deviation.

For the CAM data set, PM always used two-stage optimization. The results show that the DBAT and PM estimates of the EO, OP, and CP parameters were equal but the image residuals were not.

Finally, the S4/S5 comparison show differences at the one meter scale for the EO positions ($6.5\sigma_{\text{post}}$) and at the decimeter level for OP (around $2\sigma_{\text{post}}$).

	PM settings		$\hat{\sigma}_0$	EO diff		pos diff		
Exp.	Orient	Stages	PM/DBAT	pos	angles	OP	CP	residuals
C1A	Yes	Two	1.778/1.781	**0**	**0**	**0**	**0**	0.08
C1B	No	Two	”	**0**	**0**	**0**	**0**	”
C2A	Yes	Two	1.609/1.610	**0**	**0**	**0**	**0**	0.08
C2B	No	Two	”	**0**	**0**	**0**	**0**	”
S1A	Yes	One	0.787/1.042	**0**	**0**	-	**0**	**0**
S1B	No	One	”	**0**	**0**	-	**0**	**0**
S2A	Yes	One	0.885/0.985	0.4	0.02	-	0.02	0.25
S2B	No	One	0.984/0.985	**0**	**0**	-	**0**	**0**
S3A	Yes	One	0.871/0.965	0.33	0.01	0.017	0.021	0.25
S3B	No	One	0.965/0.965	**0**	**0**	**0**	**0**	**0**
S4A	Yes	Two	1.072/1.079	0.031	**0**	0.012	0.031	0.39
S4B	No	Two	”	”	”	”	”	”
S5	N/A		-/0.353	1.366	0.05	0.420	0.024	N/A

Table 3. Results for the experiments. The $\widehat{\sigma_0}$ estimated by PM and DBAT are given together with the maximum absolute difference between the DBAT and PM/PS estimates of the EO, OP, and CP parameters, and the image residuals. The position, angle, and residual units are meters, degrees, and pixels, respectively. All differences below 5×10^{-4} in their respective units are marked as zero and were furthermore below 0.1 times their respective posterior standard deviation σ_{post}. The non-zero values except for experiment S5 were all below $0.5\sigma_{\text{post}}$. The S5 EO differences were around $6.5\sigma_{\text{post}}$, the OP/CP differences $1.7\sigma_{\text{post}}$–$2.3\sigma_{\text{post}}$.



(a) CAM in PM

(b) CAM in DBAT

(c) SXB in PM

(d) SXB in DBAT

(e) SXB in PS

Figure 4. Reconstructed camera networks.

7. DISCUSSION

The goal of this paper was to investigate whether the Damped Bundle Adjustment Toolbox (DBAT) could be used to provide independent verification of the BA computation of PhotoModeler (PM) and PhotoScan (PS). This is a challenging task, since round-off errors, different termination criteria, etc., can generate differences that are difficult to distinguish from differences in the underlying mathematical model.

However, the results suggest that if PM performs a one-stage optimization, DBAT can replicate PM results to at least three decimals (0.1 σ_{post}), including the image residuals, for lens-distortion-free projects, with fixed or weighted control points. For fixed control points, this is true even if an orientation is performed by PM. However, if PM performs a two-stage optimization, or runs an orientation on projects with weighted control points, the results differ for unknown reasons. A speculation to be investigated is that the posterior CP standard deviations from the first stage are used as prior CP standard deviations in the second stage.

For problem with lens distortion, it appears that DBAT and PM generate the same estimates. However, the difference in image residuals indicate that the PM projection model is not completely understood.

The results furthermore suggests that PM incorrectly inflates the redundancy number r used to calculate $\widehat{\sigma_0}$ whenever fixed control points are being used. If true, the effect is largest on projects with a large number of fixed control points compared to object points. As this is the opposite of typical real-world projects, the practical impact of this difference is expected to be low.

The comparison between PM/PS is harder to do as the process also involves detection and matching of image points. Furthermore, the DBAT/PS difference is larger than the DBAT/PM difference by more than a factor of ten and significantly larger than σ_{post}, especially for the EO parameters. Part of the explanation for this is that the PS use of control points (“markers”) is not completely understood. The results do suggest that the treatment of control points is different from that of PM. This is unfortunate, as to quote Granshaw and Fraser (2015):

> Both photogrammetry and computer vision share, or at least should share, a common – or at least widely overlapping – theoretical basis.

Although the results differ, it is still possible to import PS projects into DBAT and process them as photogrammetric projects.

A conclusion that is possible to draw is that PS reports far fewer quality indicators compared to PM, such as posterior standard deviations of the estimated parameters and high correlations. DBAT reports the same quality parameters are PM.

The recent development of DBAT has enabled weighted observations to be processed. In this paper, the weighted observations were restricted to control points. However, the same code allows the toolbox to do camera self-calibration with or without weighted IO observations.

A conclusion to draw from the results reported in this paper is that if you want to trust your data, it is important to understand how your tool works. This applies to both closed and open tools. However, understanding implementation details is of course easier if the tool is open.

The use of an external verification tool such as DBAT will enable users to get an independent verification of the computations of their software. It is our hope that the availability of an external, open-source bundle verification tool will increase the quality and transparency of critical photogrammetric computations in commercial software.

ACKNOWLEDGMENTS

We acknowledge the City of Strasbourg for making the SXB images available.

References

Agisoft, 2016. Agisoft PhotoScan User Manual, Professional Edition, v1.2. St. Petersburg, Russia.

Börlin, N. and Grussenmeyer, P., 2013. Bundle adjustment with and without damping. *Photogramm Rec* 28(144), pp. 396–415.

Börlin, N. and Grussenmeyer, P., 2014. Camera calibration using the damped bundle adjustment toolbox. *ISPRS Annals of the Photogrammetry, Remote Sensing, and Spatial Information Sciences* II(5), pp. 89–96.

Brown, D. C., 1971. Close-range camera calibration. *Photogrammetric Engineering* 37(8), pp. 855–866.

Brown, D. C., 1976. The bundle adjustment — progress and prospects. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences* 21(3), pp. 33 pp.

EOS Systems, 2016. Lens distortion formulation. PhotoModeler Scanner Help System.

Granshaw, S. I. and Fraser, C. S., 2015. Editorial: Computer vision and photogrammetry: Interaction or introspection? *Photogramm Rec* 30(149), pp. 3–7.

Haralick, R. M., Lee, C.-N., Ottenberg, K. and Nölle, M., 1994. Review and analysis of solutions of the three point perspective pose estimation problem. *Int J Comp Vis* 13(3), pp. 331–356.

Hartley, R. I. and Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. 2nd edn, Cambridge University Press, ISBN: 0521540518.

Huang, T. S. and Netravali, A. N., 1994. Motion and structure from feature correspondences: A review. *P IEEE* 82(2), pp. 252–268.

Kraus, K., 1993. *Photogrammetry, Volume 1*. Ferdinand Dümmler, Bonn.

Lourakis, M. I. A. and Argyros, A. A., 2009. Sba: A software package for generic sparse bundle adjustment. *ACM TOMS* 36(1), pp. 30 pp.

Luhmann, T., Robson, S., Kyle, S. and Harley, I., 2006. *Close Range Photogrammetry*. Whittles, Scotland, UK.

McGlone, C., Mikhail, E. and Bethel, J. (eds), 2004. *Manual of Photogrammetry*. 5th edn, ASPRS.

Mikhail, E. M., Bethel, J. S. and McGlone, J. C., 2001. *Introduction to Modern Photogrammetry*. Wiley.

Tomasi, C. and Kanade, T., 1992. Shape and motion from image streams under orthography: a factorization method. *Int J Comp Vis* 9(2), pp. 137–154.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 2000. Bundle adjustment — A modern synthesis. In: *Vision Algorithms: Theory and Practice, Proceedings of the International Workshop on Vision Algorithms*, Lecture Notes in Computer Science, Vol. 1883, Spring Verlag, pp. 298–372.

Wolf, P. R. and Dewitt, B. A., 2000. *Elements of Photogrammetry*. 3 edn, McGraw-Hill, New York, USA.

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE T Pattern Anal* 22(11), pp. 1330–1334.