A two-step approach for the correction of rolling shutter distortion in UAV photogrammetry

Yilin Zhou, Mehdi Daakir, E Rupnik, Marc Pierrot-Deseilligny

Version of Record: https://www.sciencedirect.com/science/article/pii/S0924271619302849
Manuscript_38a5acc60aad3fe59c103da652eef4e8

A two-step approach for the correction of rolling shutter distortion in UAV photogrammetry

Yilin Zhou[a,*], Mehdi Daakir[b], Ewelina Rupnik[a], Marc Pierrot-Deseilligny[a]

[a] *LaSTIG, IGN, ENSG, University Paris-Est, F-94160 Saint-Mande, France*
[b] *SIXENSE Mapping, 1, Rue du docteur Charcot, 91421 Morangis, France*

Abstract

The use of consumer grade unmannaed aerial vehicles (UAV) is becoming more and more ubiquitous in photogrammetric applications. A large proportion of consumer grade UAVs are equipped with CMOS image sensor and rolling shutter. When imaging with a rolling shutter camera, the image sensor is exposed line by line, which can introduce additional distortions in image space since the UAV navigates at a relatively high speed during aerial acquisitions. In this paper, we propose (1) an approach to calibrate the readout time of rolling shutter camera, (2) a two-step method to correct the image distortion introduced by this effect. The two-step method makes assumption that during exposure, the change of camera orientation is negligible with respect to the change of camera position, which is often the case when camera is fixed on a stabilized mount. Firstly, the camera velocity is estimated from the results of an initial bundle block adjustment; then, one camera pose per scan-line of the image sensor is recovered and image observations are corrected ~~rectified~~. To evaluate the performance of the proposed method, four datasets of block and corridor configurations are acquired with the DJI Mavic 2 Pro and its original Hasselbald L1D-20c camera. The proposed method is implemented in *MicMac*, a free, open-source photogrammetric software; comparisons are carried out with other two

*Corresponding author
Email addresses: Yilin.Zhou@ensg.eu (Yilin Zhou), mehdi.daakir@sixense-group.com (Mehdi Daakir), Ewelina.Rupnik@ign.fr (Ewelina Rupnik), Marc.Pierrot-Deseilligny@ensg.eu (Marc Pierrot-Deseilligny)

Preprint submitted to ISPRS Journal of Photogrammetry and Remote Sensing, November 25, 2019

mainstream software, *AgiSoft MetaShape* and *Pix4D*, which also have the functionality of rolling shutter effect correction. For block configuration datasets, the three software give comparable results. *AgiSoft Metashape* and *Pix4D* are sensitive to the flight configuration and encounter difficulties when processing datasets in corridor configurations. The proposed method shows good robustness both in block and corridor configurations, and is the only method that works in corridor configuration. After the application of the rolling shutter effect correction, the 3D accuracy is improved by 30%-60% in block configuration and 15%-25% in corridor configuration. A further improvement can be expected if a precise dating of image is available or if the camera positions can be directly extracted from GNSS data.

Keywords: Rolling shutter, readout time calibration, distortion correction, UAV, aerial photogrammetry.

2019 MSC: 00-01, 99-00

---

1. Introduction

The use of consumer grade unmanned aerial vehicles (UAV) for photogrammetric applications is becoming increasingly ubiquitous thanks to its low cost, good portability, ease of use and high flexibility (Nex and Remondino (2014); Vautherin et al. (2016)).

A great variety of consumer and professional cameras can be found on the market nowadays. Digital single-lens reflex cameras (DSLRs) have replaced film-based SLRs and remain the most popular type of interchangeable lens camera. There are two main categories of image sensor, complementary metal-oxide semiconductor (CMOS) sensor and charge-couple device (CCD) sensor. CMOS sensors' lower cost and lower power consumption compared to CCD sensors make them a common choice for consumer and pro-sumer cameras. The vast majority of CCD sensors are equipped with global shutter whereas CMOS sensors often employ rolling shutters.

Despite the fact that certain UAV manufacturers propose UAVs equipped

with global shutter cameras, a large proportion is still equipped with rolling shutter complementary metal oxide semiconductor (CMOS) sensors. In a rolling shutter readout, not every portion of the image sensor is exposed at the same time but in a progressive motion as in traditionnal dual curtain focal-plane shutters. Additional distortion in image space can be found when the object or the image sensor is moving during the exposure, which is normally the case for aerial photogrammetric acquisitions. Without proper modeling and processing, the rolling shutter effect can degrade the accuracy of photogrammetric reconstruction by a wide margin.

In this article, a two-step rolling shutter correction method is presented: we assume that the rotational velocity is negligible compared to the translational velocity. This assumption is adopted for the following reasons: (1) the simulation results in Zhou et al. (2019) confirm this assumption; (2) the camera rotation is often diminished by for instance, a stabilized mount, as in our experiment; (3) we don't really have access to angular information in our experiment. The camera translational velocity is estimated with time information in image metadata and camera positions issued from an initial bundle block adjustment; the camera positions can also be extracted when GNSS data are available. Afterwards, the camera position at exposure of each line is recovered and the observations in image space are corrected ~~rectified~~ accordingly. This method is implemented in *MicMac*, a free, open-source photogrammetric software (Rupnik et al. (2017)). To evaluate the performance of the proposed method, four datasets of block and corridor configurations are processed without and with the rolling shutter correction. Comparisons with two mainstream photogrammetric software: *AgiSoft MetaShape* and *Pix4D* are carried out with the same datasets and the same configurations.

2. State of the art

A great deal of studies have been conducted on the rolling shutter effect by the photogrammetry and computer vision communities. In Ait-Aider et al.

(2006), the image deformation introduced by rolling shutter is leveraged for the estimation of instantaneous 3D pose and 3D velocity. The proposed method assumes the object motion to be uniform during the image exposure. It gives good 3D pose estimation in non-planar cases whereas for planar cases the accuracy is less satisfying. A generic camera model is given in Magerand and Bartoli (2010) for handling both uniform and non-uniform camera motions. A concept of dynamic pose is proposed, similarly to a push-broom camera, the pose is written as a function of the scan-line being considered. A projective geometry of the rolling shutter camera and how it is affected by different types of camera motion are presented in Meingast et al. (2005). In the case of fronto-parallel motion, the camera is shown to be modeled as an X-slit-camera.

Substantial researches are carried out to compensate the rolling shutter effect in image space. These methods can be refered to as **two-step** methods, since they first correct image measurements before performing the photogrammetric processing. In Liang et al. (2008), an image space analysis of the rolling shutter effect is described, the compensation of the distortion is performed with global planar motion estimation, parametric curve fitting and scanline realignment. By assuming the motion to be smooth and performing curve interpolation for each pixel, the rolling shutter distortion in videos is corrected ~~rectified~~ in Sun and Liu (2012). Klein and Murray (2009) estimates velocities for each image from adjacent images and compensate the tie points. Chun et al. (2008) described the rolling shutter distortion in terms of 2D velocity and implemented an affine transformation correction. Sun et al. (2016) computed a sequence of affine transformation represented by a multilayer neural network for the compensation of rolling shutter effect. A mixture of homographies is explored for a calibration-free rolling shutter removal in Grundmann et al. (2012). The method proposed in Ito and Okatani (2017) assumes a linear rotation only camera motion, the problem is recast as camera self-calibration with a varying skew and aspect ratio for each image sequence.

Focuses are also given on rolling shutter bundle adjustments. These methods are refered to as **one-step** methods, the rolling shutter correction is per-

formed simultaneously within the bundle block adjustment. Hedborg et al. (2011) demonstrated a structure and motion scheme for rolling shutter videos which works with general camera motions. Saurer et al. (2016) adopts a constant translational and rotational velocity parametrization for the camera pose and proposes a cost function for Bundle Adjustment (BA) that models the rolling shutter effect. *Pix4D* assumes as well a constant camera motion and describes the rolling shutter effect with a 6-parameter model (Vautherin et al. (2016)). As for *Agisoft MetaShape*, its rolling shutter correction method is not presented publicly.

The method proposed in this article is a **two-step** approach. It performs image measurement correction independently from the photogrammetric processing, which makes it more robust to image block geometry. Unlike most **two-step** methods, the approach presented here explores the image block geometry for the camera motion estimation and the image measurement correction.

3. Problem formulation

For modern camera equipped with a rolling shutter, the image sensor is activated and read out one row at a time. The exposure takes place between the time the first row is exposed and the time the last row is read out. The readout time for each frame is constant and does not vary with exposure parameters. In a word, an important delay due to the readout time can be found between exposures of the first and the last row, which introduces distortion when imaging a mobile object or the image sensor itself moves during exposure. Figure 1 gives an illustration of the general rolling shutter scheme.

In the case of aerial photogrammetric acquisition, the speed of an unmanned aerial vehicle (UAV) embarking with optical sensors can reach 10 m/s. Typically, the readout time of consumer grade cameras varies in the range of 30 ms to 80 ms, which results in a non-negligible camera position change between the beginning and the end of exposure.

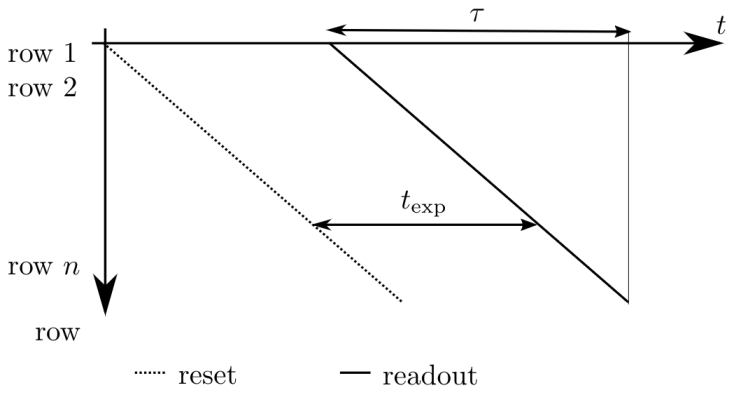


Figure 1: Rolling shutter readout scheme. The sensor is exposed row by row at a constant speed. After the exposure duration t_{exp} , the sensor starts the readout row by row. At time t= 0 the exposure of the first row takes place. It is then read out at time $t = t_{exp}$. Consecutive rows are exposed and read out one after the other. The sensor readout is finished after the rolling shutter readout time τ. (source: Vautherin et al. (2016))

4. System and calibration

4.1. Drone and camera

For the study of rolling shutter effect as well as the aerial data acquisition, the DJI Mavic 2 pro (Figure 2) and its original camera, a Hasselblad L1D-20c camera are employed. The camera is equiped a 3-axis gimbal to provide a stable, smooth footage for image acquisition. The camera specifications are given in Table 1:

Figure 2: DJI Mavic 2 Pro

Sensor	1″ CMOS Effective Pixels: 20M
Lens	35 mm Format Equivalent: 28mm
Still image size	5472 × 3648

Table 1: Camera specifications (`https://www.dji.com/uk/mavic-2`)

4.2. Calibration of readout time

No specifications of the rolling shutter readout time are given on the website of DJI. It occurs sometimes that this information remains absent and is not given by camera manufacturers since it may be viewed as a camera defect. In one blog of *Pix4D*, the readout time of 6 DJI models are listed[1]. However the model employed for our research, DJI Mavic 2 Pro, is not included. Therefore, we propose here an approach to calibrate the rolling shutter readout time for the following reasons: (1) the readout time is often unprovided by the camera manufacturer; (2) there are few methods proposed by the community to calibrate the readout time; (3) it is more rigorous to have a control on the readout time even when it is given; (4) from an educational point of view, it is a good exercise for students to understand the rolling shutter effect.

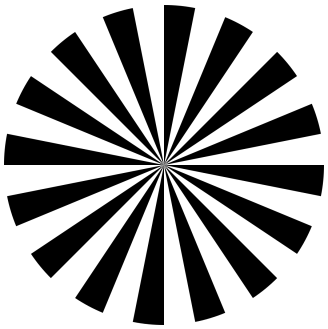
System setup.

The setup of the readout time calibration system is shown in Figure 3: the drone is placed in front of a Siemens star with the camera facing front. The Siemens star consists of a set of spokes radiating from a common center which become wider as they get further[2]. It is connected to a motor and can rotate in a constant speed. A magnet is fixed on the back of the Siemens star. The time interval between its two passages is detected by a magnetic field sensor, the instantaneous angular frequency is then calculated and displayed on a screen at bottom left (see Figure 3a).

[1] `https://www.pix4d.com/blog/rolling-shutter-correction`

[2] `https://en.wikipedia.org/wiki/Siemens_star`

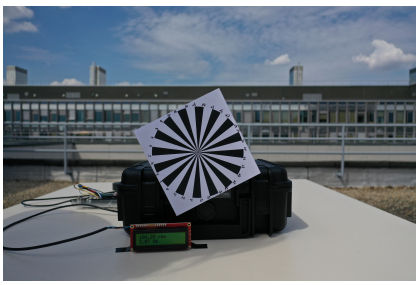
(a) The readout time calibration system



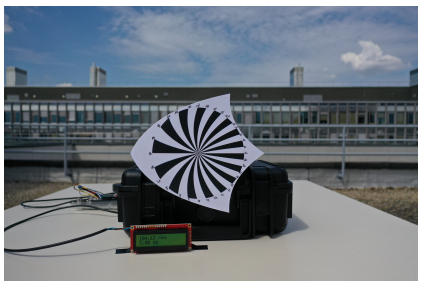
(b) The Siemens star

Figure 3: An illustration of the system setup for readout time calibration.

Two sets of photos are acquired with DJI Mavic 2 Pro; the first set is acquired with the Siemens star being static, the second one with the Siemens star rotating in a certain angular speed. The photographic parameters are configured identically, the shutter speed is set to 1/4000 s so that the exposure time t_{exp} is negligible with respect to the readout time τ. One example photo of each set is given in figure 4.



(a) Image taken with static object



(b) Image taken with rotating object

Figure 4: An example of images taken with a rolling shutter camera.

Mathematical model.

The mathematical model for the estimation of readout time τ is illustrated in Figure 5. For each spoke of the Siemens star, two points A_1, A_2 can be identified and share the same distance R to the center O. The image sensor of

in total L rows is exposed and read out from top to bottom with a readout time τ. The Siemens star rotates counterclockwise with a constant angular velocity ω.

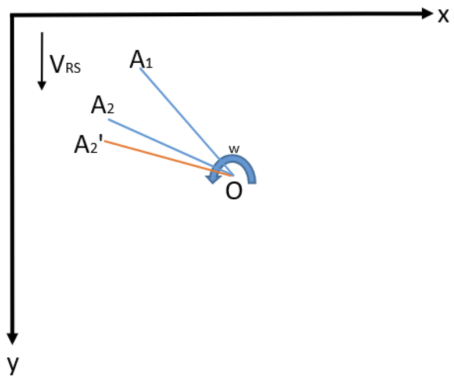


Figure 5: Illustration of the mathematical model for the estimation of readout time.

Assume that at $t = t_1$, the row y_{A_1} of the image sensor is exposed and read out; the point A_2 has not been imaged yet. At $t = t_2$, the point A_2 moves to point A_2' and the row $y_{A_2'}$ of the image sensor is exposed. The time interval between these two exposures can be expressed by:

$$t_2 - t_1 = \frac{y_{A_2'} - y_{A_1}}{L} \cdot \tau \tag{1}$$

Note the angles $\theta = \angle A_1 O A_2$ and $\theta' = \angle A_1 O A_2'$, the time interval $t_2 - t_1$ can also be expressed by:

$$t_2 - t_1 = (\theta' - \theta)/\omega \tag{2}$$

The two angles θ and θ' have the following relations:

$$\cos(\frac{\pi - \theta}{2}) = \frac{\overline{A_1 A_2}}{2R}, \quad \theta = \pi - 2 \cdot \arccos(\frac{\overline{A_1 A_2}}{2R}) \tag{3}$$

$$\cos(\frac{\pi - \theta'}{2}) = \frac{\overline{A_1 A_2}}{2R}, \quad \theta' = \pi - 2 \cdot \arccos(\frac{\overline{A_1 A_2}}{2R}) \tag{4}$$

Given (1) and (2), one can obtain:

$$\tau = \frac{L \cdot (\theta' - \theta)}{\omega \cdot (y_{A_2'} - y_{A_1})} \tag{5}$$

By replacing θ and θ' with (3) and (4), the readout time:

$$\tau = \frac{2L \cdot [\arccos(\frac{\overline{A_1 A_2}}{2R}) - \arccos(\frac{\overline{A_1 A_2'}}{2R})]}{\omega \cdot (y_{A_2'} - y_{A_1})} \tag{6}$$

To generalize,

$$\tau = \frac{L \cdot |\theta' - \theta|}{\omega \cdot |y_{A_2'} - y_{A_1}|} \tag{7}$$

Calibration results.

For each spoke of the Siemens star, two points are identified and manually measured, which adds up to 32 points in one image. Figure 6 shows the positions of these 32 points in the above-mentioned two cases: the Siemens star remains static (left) and the Siemens star rotates (right), respectively. One can observe that, in the static case, the 32 points are equally separated as in object space. As for the case in which the Siemens star rotates counterclockwise and the image sensor is exposed from top to bottom, the distances between points on the left side are bigger than the ones on the right side. For points on the right side which ascend during the exposure, by the time the exposure moves to the next rows, points which were at a lower part of the image catch up and get to be exposed, hence a shorter distance between points.

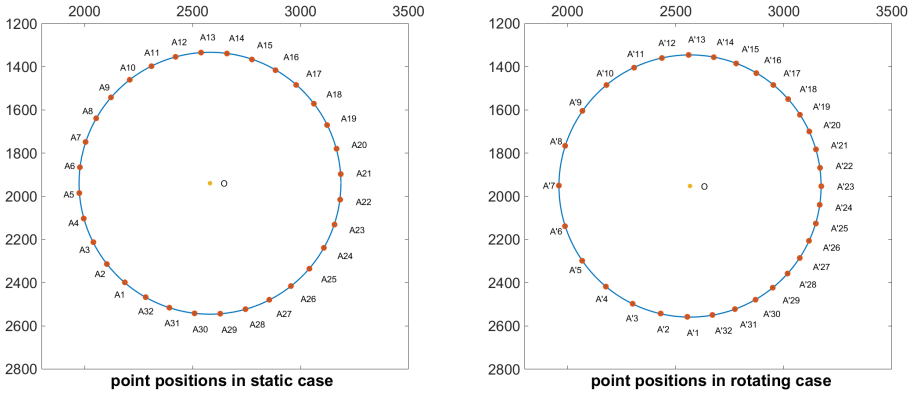


Figure 6: Point positions in images, image sensor is exposed from top to bottom in image space. **Left**: static case. **Right**: rotating case.

A total of 154 point couples are extracted from the static/rotating image pair. Selections are made so that the two points involved for estimation are on

the same side of the circle (e.g., pure descending or ascending during imaging). A linear least squares fitting is performed with the *Curve Fitting Toolbox* in *Matlab*. Figure 7 demonstrates the fitting results. The estimated readout time $\tau = 56.4$ms with the R-squared (coefficient of determination) equals to 0.9997.

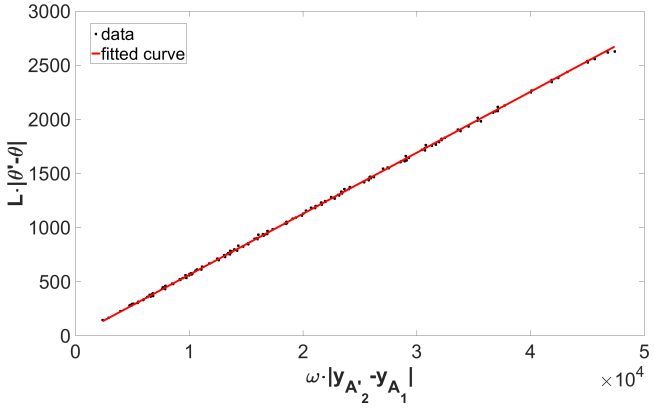


Figure 7: Linear fitting results for readout time estimation.

5. Research design

To improve the accuracy of photogrammetric results, the following pipeline is proposed to correct the rolling shutter effect on images. Note that the assumption is made that there is no variation of camera orientation during exposure. Firstly, the simulation results in Zhou et al. (2019) showed that the impact of angular motion is negligible compared to that of translational motion. Moreover, the camera employed here is equipped with a 3-axis gimbal for a stable imaging. The proposed method is a two-step solution which in the first instance estimates the camera translational velocity from an initial bundle block adjustment result, then recovers the camera position for each line at exposure to rectify image measurements. The implementation is carried out as follows:

Step 1: obtain an initial solution.

A standard photogrammetric pipeline is performed to obtain an initial solution. It consists of identifying common feature points between overlapping

images and recovering their poses firstly in a relative coordinate system. The geo-referencing and a second bundle block adjustment in absolute scale are carried out with the help of ground control points (GCPs).

Step 2: calculate image sensor velocity.

In our case with DJI Mavic 2 Pro, the time related information of the exposure is recorded in the metadata of each image. An initial solution of the camera position and orientation is issued after *step 1*. The velocity of one image is considered as the ratio between the displacement and the time interval of its two adjacent images. Note that the time related information in the metadata of one image is often rounded in second. Therefore, there is no much interest in applying a more sophisticated approach for the estimation of image sensor velocity.

We observed that the DJI Mavic 2 Pro gives erroneous GNSS measurements and the issue has not been fixed until after the acquisitions are conducted[3]. Therefore, the GNSS data are not explored. In the case where correct GNSS measurements are available, the camera velocity can be estimated more rigorously with the help of time information in image metadata and camera positions extracted from GNSS data; the *Step 1* can then be skipped.

Step 3: recover camera position at exposure.

For image i, its camera position $C_i = [C_{i,x}, C_{i,y}, C_{i,z}]$ is acknowledged with *Step 1*. In the case where the moving direction of the drone and the that of the rolling shutter are opposite (as in Figure 4), the acquired images are dilated compared to images without rolling shutter effect ~~the ones obtained with global shutter~~. ~~For an image sensor with L rows, the camera position C_i obtained with *Step 1* is closer to the instant camera position when the middle row $\frac{L}{2}$ of the image sensor is exposed.~~ The camera position C_i obtained with *Step 1* is closer to the instantaneous camera position when the middle part of the image sensor is exposed. For an image sensor of L rows, this position can be noted

[3] `https://forum.dji.com/thread-186813-1-1.html`

as $C_i^{\frac{L}{2}}$. For row l, the time elapsed between when it is exposed and when the middle row is exposed can be calculated once the rolling shutter readout time τ is known. The readout time being small enough (less than 100 ms), the image sensor velocity V_{uav} is assumed to be constant during the readout phase and is calculated with *Step 2*. The camera position C_i^l when the row l is exposed can be expressed then by:

$$C_i^l = C_i + V_{uav} \cdot \tau \cdot \frac{l - \frac{L}{2}}{L} \tag{8}$$

In a nutshell, from one camera position C_i, a set of camera positions $\left[C_i^0, \cdots, C_i^L\right]$ corresponding to row $[0, \cdots, L]$ are recovered.

Step 4: correct image measurements.

A set of images see a 3D point at $\mathbf{p} = [p_1, \cdots, p_n]^T$. The 2D observations $\mathbf{p}$ are extracted with feature detection algorithm (e.g. SIFT), the corresponding camera poses $\mathbf{\Omega} = [\Omega_1, \cdots, \Omega_n]^T$ are issued from *Step 1*.

Firstly, the 3D coordinates p of the point are recovered by performing a pseudo-intersection Cheng et al. (2001) $\psi(.) : \mathbb{P}^2 \rightarrow \mathbb{P}^3$:

$$P = \psi(\mathbf{\Omega}, \mathbf{p}) \tag{9}$$

Then, the 3D point is reprojected on images with the projection function $\pi(.) : \mathbb{P}^3 \rightarrow \mathbb{P}^2$, which gives image measurements $\mathbf{p}' = [p'_1, \cdots, p'_n]$.

$$\mathbf{p}' = \pi(\mathbf{\Omega}, P) \tag{10}$$

The differences between these two sets of image measurements comes from random and systematic errors of feature detection and camera internal/external orientation calculation.

$$\delta\mathbf{p} = \mathbf{p}' - \mathbf{p} \tag{11}$$

A second reprojection is conducted with the corrected camera positions $\mathbf{\Omega}_c$ issued from *Step 3*,

$$\mathbf{p}'_c = \pi(\mathbf{\Omega}_c, P) \tag{12}$$

Then the obtained image measurements $\mathbf{p}'_c$ are corrected ~~rectified~~ with image measurement residuals $\delta\mathbf{p}$,

$$\mathbf{p}_c = \pi(\mathbf{\Omega}_c, P) - \delta\mathbf{p} \tag{13}$$

The final image measurements $\mathbf{p}_c$ are considered the corrected observations. Figure 8 presents the relation between different image measurements mentioned above.

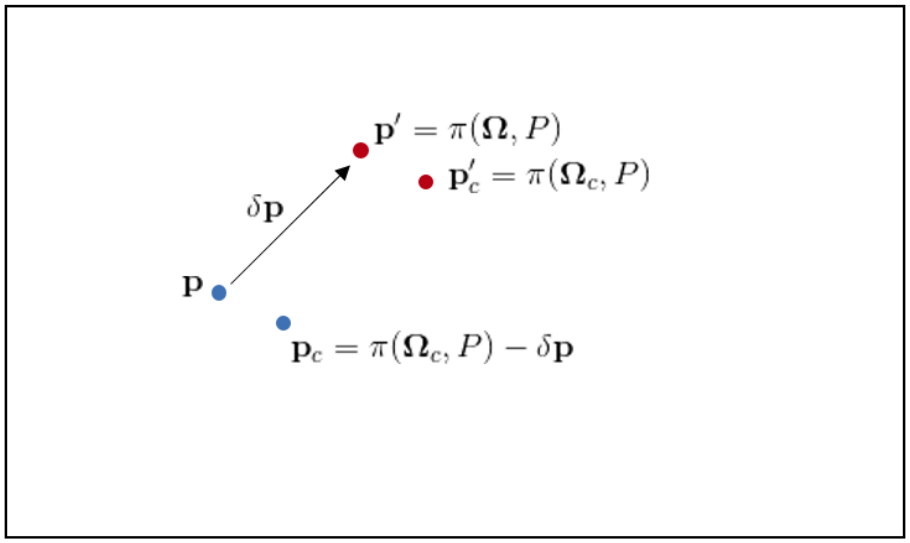


Figure 8: Illustration of different image measurements mentioned in **Step 4**. P and P_c correspond to the original and the corrected image measurements, respectively.

6. Experiments and results

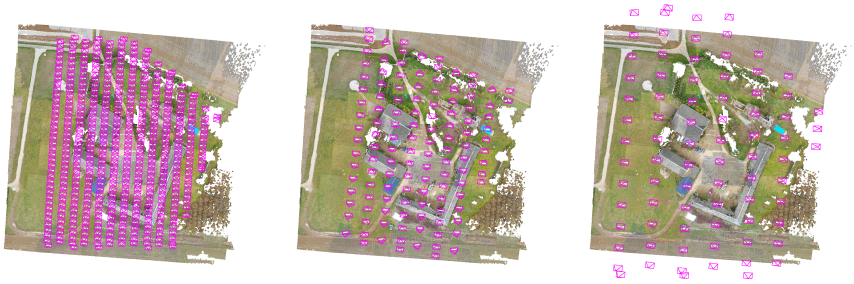
6.1. Introduction of datasets

The aerial acquisition is conducted on an aerodrome at Viabon, France. Figure 9 shows two zones that have been surveyed with the DJI Mavic 2 Pro. One zone in block configuration and one in corridor configuration, respectively.

The block configuration zone is about 150m × 200m and 3 flights are carried out in this area: two flights at the height of 30m and 90m with nadir-looking

Figure 9: Illustration of the two surveyed zones in block and corridor configurations.

camera and one flight at the height of 60m with oblique-looking camera. The corridor area extends for 400m's long and is surveyed with one nadir-looking flight at the height of 40m. Figure 10 and Table 2 gives more information about the conducted flights.



(a) nadir flight at 30m (b) Oblique flight at 60m (c) nadir flight at 90m

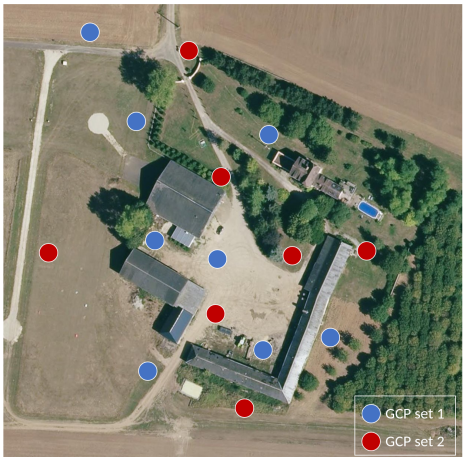
(d) nadir flight at 40m

Figure 10: An illustration of the conducted flights.

Flight	naidr-30m	oblique-60m	nadir-90m	nadir-40m
Nb of images	443	125	80	136
Overlap (%)		80/80		
GSD(cm)	0.74	∼1.48	2.10	0.96

Table 2: Details of conducted flights

A total of 23 ground points are surveyed with Leica 2000 receivers in D-RTK mode, 15 of which are visible in the block configuration datasets, and 11 in the corridor configuration dataset (3 points are seen by both configurations). For each configuration, the seen ground points are divided into two sets, one serves as ground control points (GCP) and the other as check points (CP). Figure 11 shows the distribution of GCPs and how they are divided into two sets.



(a) block configuration

(b) corridor configuration

Figure 11: Distribution of GCPs

6.2. Camera calibration model

Two camera calibration models are employed for the photogrammetric processing, each estimates 8 and 10 parameters respectively.

The **8-parameter** calibration model consists of

- f: the camera focal length
- PP_x, PP_y: the principal point, which is also the distortion center
- K_1, K_2, K_3: third-order ~~3 degrees of~~ radial distortion coefficients
- P_1, P_2: decentring distortion coefficients

The **10-parameter** calibration model includes in addition

- b_1, b_2: the affine distortion coefficients

The 10-parameter camera model is actually the well-known *Fraser* camera model (Fraser (1997)). The intention of employing these 2 camera models and investigating the influence of the affine distortion coefficients b_1, b_2 is that, in the case of a flat scene as in aerial acquisitions, the rolling shutter effect may cause a similar effect to the affine distortion which is supposed to be corrected by b_1, b_2. Therefore, it is interesting to see until which point the affine distortion coefficients can correct the rolling shutter effect as well as if the proposed method can bring a further improvement.

6.3. Data processing

The same data processing pipeline is performed on the original and the corrected dataset. For the corrected dataset, the tie points and the image measurements are modified as explained in Section 5 while the rest remains unchanged with respect to the original dataset.

Firstly, a bundle block adjustment is performed with the tie points as input to recover the camera poses in a relative coordinate system. Afterwards, GCPs are used to geo-reference the 3D model and to perform a second bundle block adjustment in an absolute reference frame. It is worth noting that each time

there is one set of points that participate in the bundle block adjustment as GCPs, the other set is used as CPs to evaluate the accuracy of the issued results. Then the same procedure is carried out with GCPs and CPs being swapped. It allows to verify the consistency of the obtained results and to eliminate the occasionality of GCP distribution.

To summarize, for one dataset, the original and the corrected observations are explored; two camera models are employed for processing; two combinations of GCPs/CPs are used. Therefore, each dataset is processed in 8 configurations. The notation of the configurations is given in Table 3:

		Original	Corrected
8-param	GCP set 1	o-8p-GCP1	c-8p-GCP1
	GCP set 2	o-8p-GCP2	c-8p-GCP2
10-param	GCP set 1	o-10p-GCP1	c-10p-GCP1
	GCP set 2	o-10p-GCP2	c-10p-GCP2

Table 3: Notation of configurations; "o" and "c" for original and corrected observations; "8p" and "10p" for 8-parameter and 10-parameter camera models; "GCP1" and "GCP2" for point set 1 and 2 as GCPs.

We choose four datasets to be processed:

1. nadir-30m, block configuration
2. nadir-90m, block configuration
3. nadir-30m + nadir-90m + oblique-60m, block configuration
4. nadir-40m, corridor configuration

The comparison between (1) and (2) gives a hint on how the flight height impacts the accuracy. By comparing (1) (2) with (3), the influence of flight configuration (i.e. inclusion of oblique images and nadir images taken from different flight heights) can be investigated. The (4) allows to verify if the rolling shutter correction works as well in corridor configurations.

6.4. Results

The proposed method for rolling shutter effect correction is implemented in *MicMac* [28], a free, open-source photogrammetric software developed at IGN (French Mapping Agency) and ENSG (Ecole Nationale des Sciences Géographiques) since 2003 (Rupnik et al. (2017)). Two well-known commercial photogrammetric software, *AgiSoft MetaShape* and *Pix4D*, also have the functionality of rolling shutter correction. Therefore, the same datasets are processed with all these three software in the same configuration and the performance of the rolling shutter correction are compared and analyzed.

6.4.1. MicMac

It is worth bearing in mind that, a tie point reduction is performed on SIFT-extracted tie points in *MicMac* to reduce memory requirements and processing time while maintaining the tie point manifold (Martinez-Rubi (2016)). Figure 12 gives histograms on the multiplicity for SIFT-extracted tie points with *MicMac* before and after the tie point reduction. One can see that the reduction procedure lowers the number of tie points as well as the percentage of tie points with poor multiplicities. Hence, the data processing requires less compute capacity and time while still offering a guaranteed performance.

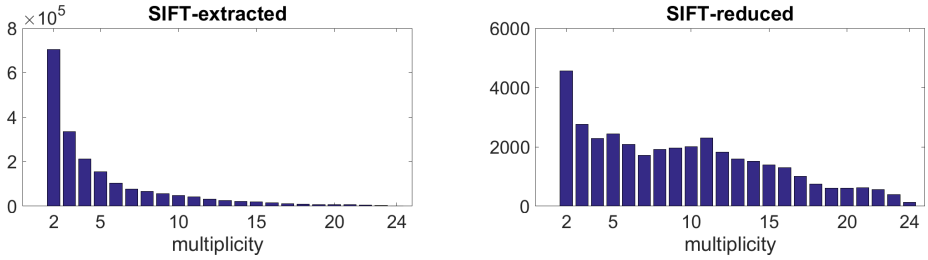


Figure 12: Histogram of tie point multiplicity before and after the tie point reduction.

The following Tables 4-7 give data processing results of the proposed approach. The ~~results are presented with~~ modeling accuracy is evaluated with the errors on check points and is given in root-mean-square error (RMSE), mean and unbiased standard deviation (STD) in planimetry, altimetry and 3D, re-

spectively. The improvement after applying the rolling shutter correction is calculated based on the variation of 3D RMSE. The best configuration for each dataset is highlighted in red.

Table 4, 5: For dataset 30m-nadir and 90m-nadir in block configuration, the addition of affine distortion coefficients improves the accuracy in planimetry, especially with the original datasets. The planimetric accuracy is further improved after the rolling shutter effect correction. The best results are obtained after the correction of rolling shutter effect, both camera calibration models give comparable result with the 10-parameter model outperforming slightly the 8-parameter one. The dataset 2 (90m-nadir) gives better results compared to the dataset 1 (30m-nadir) when the rolling shutter effect is corrected, which is probably due to the fact that the results are less sensitive to the estimation error of camera velocity (due to rounded time information) with a higher surveying speed.

Table 6: the dataset 3 shows similar results to dataset 1 and 2 in the cases of original datasets. In the cases where the rolling shutter effect is corrected, the inclusion of affine distortion parameters degrades the planimetric accuracy. This is probably due to the fact that, with a proper flight configuration (oblique images and nadir images taken from different flight heights) being present, the rolling shutter effect has been corrected almost entirely by the proposed method and the dataset is over-parameterized with the 10-parameter model.

Table 7: the dataset 4 shows similar results to the dataset 3. However, the improvements are less significant in corridor configuration (15% - 25%) compared to in block configurations (30% - 60%). With the corridor configuration being an unfavorable flight configuration, a bowl effect is often observed and aggravates the obtained accuracy, even for cases in which no rolling shutter effect is present. Therefore, the improvement in this configuration is less significant compared to block configurations.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	10.6	9.4	5.3	6.9	-1.7	7.3	12.6	11.6	5.4	c-8p-GCP1	3.3	2.9	1.6	6.9	-1.6	7.4	7.7	6.9	3.8	69	0
o-8p-GCP2	14.7	12.6	8.1	10.3	-2.8	10.6	18.0	15.1	10.3	c-8p-GCP2	4.7	2.8	4.0	6.4	4.0	5.3	7.9	6.3	5.2	68	38
o-10p-GCP1	3.7	3.3	1.8	6.2	-1.6	6.6	7.3	6.8	2.8	c-10p-GCP1	2.5	2.2	1.3	6.9	-1.5	7.4	7.4	6.5	3.8	32	-11
o-10p-GCP2	6.6	5.7	3.6	10.0	-2.7	10.3	12.0	9.4	7.9	c-10p-GCP2	4.1	2.7	3.3	6.3	4.0	5.2	7.6	6.1	4.8	38	37

Table 4: *MicMac* - Results for dataset 1: 30m-nadir.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	14.1	12.8	6.4	4.0	-1.6	4.0	14.6	13.5	6.1	c-8p-GCP1	3.1	2.5	1.9	4.9	0.4	5.3	5.8	4.9	3.4	78	-22
o-8p-GCP2	13.9	12.5	6.4	5.2	-0.3	5.6	14.8	13.5	6.6	c-8p-GCP2	2.8	1.8	2.2	4.8	-1.0	5.0	5.6	3.9	4.3	80	8
o-10p-GCP1	6.3	5.9	2.5	4.0	-1.2	4.1	7.5	7.1	2.5	c-10p-GCP1	1.9	1.5	1.1	4.9	0.4	5.2	5.2	4.2	3.3	69	-22
o-10p-GCP2	7.8	6.6	4.3	5.8	-0.1	6.2	9.7	8.3	5.3	c-10p-GCP2	2.0	1.5	1.5	4.7	-1.1	4.9	5.1	3.7	3.8	74	19

Table 5: *MicMac* - Results for dataset 2: 90m-naidr.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	14.0	12.8	6.1	3.2	0.0	3.5	14.4	13.3	5.9	c-8p-GCP1	4.1	3.7	1.9	2.4	1.6	1.9	4.8	4.4	1.9	70	25
o-8p-GCP2	13.3	11.9	6.1	3.0	-0.3	3.2	13.6	12.6	5.6	c-8p-GCP2	4.4	3.1	3.4	2.2	-1.9	1.1	5.0	4.1	3.0	67	27
o-10p-GCP1	9.0	8.1	4.2	3.4	0.0	3.6	9.6	8.9	3.9	c-10p-GCP1	5.8	5.0	3.2	2.3	1.5	1.9	6.3	5.7	2.8	36	32
o-10p-GCP2	8.7	7.5	4.6	3.1	-0.2	3.3	9.3	8.4	4.1	c-10p-GCP2	5.9	4.1	4.5	2.1	-1.9	1.1	6.3	4.9	4.2	32	32

Table 6: *MicMac* - Results for dataset 3: 30m-nadir + 60m-oblique + 90m-nadir.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	6.4	5.2	4.2	8.1	-2.9	8.3	10.4	8.7	6.2	c-8p-GCP1	5.3	4.0	3.8	5.9	-1.8	6.2	8.0	6.6	4.9	17	27
o-8p-GCP2	6.4	4.7	4.9	11.7	6.0	11.3	13.4	8.4	11.7	c-8p-GCP2	1.6	1.5	0.8	9.9	4.4	9.8	10.0	6.0	8.9	75	15
o-10p-GCP1	5.3	4.3	3.3	7.7	-2.8	7.8	9.4	7.8	5.6	c-10p-GCP1	5.1	4.1	3.4	6.3	-1.9	6.6	8.1	6.9	4.6	4	18
o-10p-GCP2	5.0	3.7	3.8	11.5	5.8	11.1	12.6	7.6	11.2	c-10p-GCP2	3.2	2.8	1.7	10.0	4.6	9.9	10.5	6.5	9.2	36	13

Table 7: *MicMac* - Results for dataset 4: 40m-nadir.

6.4.2. AgiSoft MetaShape

There is not much information on how the rolling shutter correction is implemented in *AgiSoft MetaShape*. What is known is that the software gives information on rotational and translational velocity for every image. However, it is not clear if the parameters are estimated during the bundle block adjustment or calculated *a posteriori*.

Table 8, 9: in the case where no rolling shutter correction is performed, the inclusion of affine distortion coefficients improves substantially the planimetric accuracy. The correction of rolling shutter effect does improve the results when processed with the 8-parameter camera model; the error on planimetry is reduced considerably whereas on altimetry the error increases slightly. The best

results are obtained in the cases where the original datasets are processed with the 10-parameter camera model, whereas the correction of rolling shutter effect degrades the accuracy.

Table 10: With a proper flight configuration, the correction of rolling shutter effect improves significantly the obtained accuracy for both camera models. The 10-parameter camera model achieves an accuracy slightly better then that of the 8-parameter camera model.

Table 11: When processing the datasets without the rolling shutter correction, the obtained accuracy is satisfying. However, the rolling shutter correction encounters difficulties when it comes to corridor configuration; the accuracy degrades drastically, mostly in altimetry. Though the method used by *MetaShape* is not documented, one can infer that the camera motion is estimated within the bundle block adjustment with supplementary parameters, as in *Pix4D*. Therefore, when the flight configuration is not optimal, for instance, the corridor configuration, the system is over-parameterized. Thus, the application of rolling shutter correction degrades the results.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	10.5	9.7	4.5	3.2	0.4	3.4	11.5	10.4	5.5	c-8p-GCP1	0.5	0.4	0.2	7.4	-3.4	7.2	7.6	4.6	6.7	95	-131
o-8p-GCP2	8.3	8.2	4.7	15.4	-0.4	16.4	19.8	17.2	7.2	c-8p-GCP2	0.2	0.2	0.1	8.1	3.7	7.7	8.4	6.3	6.3	98	47
o-10p-GCP1	1.2	1.2	0.4	2.5	0.4	2.7	2.8	2.4	1.6	c-10p-GCP1	1.8	1.4	1.2	6.9	-3.1	6.8	7.2	4.2	6.4	-50	-176
o-10p-GCP2	3.5	3.0	1.8	3.2	0.8	3.3	4.8	4.1	2.5	c-10p-GCP2	2.2	2.1	1.2	8.2	3.8	7.9	8.5	6.5	6.4	37	-156

Table 8: *MetaShape* - Results for dataset 1: 30m-nadir.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	15.3	13.3	8.1	3.5	1.0	3.6	15.7	13.9	7.9	c-8p-GCP1	2.6	2.1	1.6	4.9	-0.2	5.3	5.6	4.2	3.9	82	-40
o-8p-GCP2	14.8	11.2	7.4	8.0	0.4	7.9	16.9	11.5	9.5	c-8p-GCP2	2.1	1.5	1.2	13.9	0.3	14.1	14.0	7.4	10.0	85	-74
o-10p-GCP1	4.8	4.4	2.0	2.4	0.7	2.5	5.4	5.1	1.6	c-10p-GCP1	3.2	2.6	2.0	7.0	1.3	7.4	7.7	5.6	5.7	33	-229
o-10p-GCP2	6.0	4.1	3.3	5.4	-1.5	5.8	8.1	4.6	5.3	c-10p-GCP2	2.9	2.0	1.7	14.9	-0.3	15.5	15.2	9.0	10.2	52	-176

Table 9: *MetaShape* - Results for dataset 2: 90m-naidr

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	14.7	13.3	6.6	4.1	-1.5	4.2	15.2	13.9	6.6	c-8p-GCP1	2.2	1.9	1.1	1.5	-1.0	1.2	2.6	2.4	1.1	85	63
o-8p-GCP2	13.4	11.0	6.0	3.6	2.0	3.0	13.9	11.8	5.3	c-8p-GCP2	2.8	1.8	1.8	2.4	0.7	2.4	3.7	3.0	1.7	79	33
o-10p-GCP1	8.7	7.9	4.1	4.4	-1.2	4.6	9.8	9.0	4.1	c-10p-GCP1	2.0	1.8	0.9	1.5	-1.0	1.3	2.5	2.4	1.0	77	66
o-10p-GCP2	7.7	5.2	4.4	3.3	1.6	3.0	8.4	6.4	3.8	c-10p-GCP2	2.3	1.4	1.6	2.5	0.7	2.6	3.4	2.9	1.6	70	24

Table 10: *MetaShape* - Results for dataset 3: 30m-nadir + 60m-oblique + 90m-nadir.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	5.7	4.1	4.4	4.7	1.4	4.9	7.4	5.6	5.4	c-8p-GCP1	10.6	7.5	8.2	14.2	7.5	13.2	17.7	14.8	10.7	-86	-202
o-8p-GCP2	5.9	4.1	4.7	5.1	-1.8	5.4	7.8	5.2	6.5	c-8p-GCP2	8.3	7.7	3.3	45.0	-22.0	43.9	45.8	30.5	38.1	-40	-782
o-10p-GCP1	4.5	3.1	3.6	4.4	1.1	4.6	6.3	4.7	4.6	c-10p-GCP1	17.1	10.9	14.4	56.8	6.7	61.8	59.3	47.0	39.7	-280	-119
o-10p-GCP2	4.8	3.3	3.8	5.3	-1.8	5.6	7.1	4.7	6.0	c-10p-GCP2	31.9	29.7	13.2	91.2	-41.7	90.7	96.6	70.3	74.1	-564	-1620

Table 11: *MetaShape* - Results for dataset 4: 40m-nadir.

6.4.3. Pix4D

In Vautherin et al. (2016) it is shown that a linear motion assumption is made. The motion velocity is described with a 6-parameter model $[\Delta R_x, \Delta R_y, \Delta R_z, \Delta T_x, \Delta T_y, \Delta T_z]$. There is one thing to point out: in *Pix4D* solely the 8-parameter camera calibration model is supported. Therefore, only 4 configurations concerning the 8-parameter camera model are conducted.

Table 12, 13, 14: all three datasets show significant increases on accuracy after applying the rolling shutter correction. With no rolling shutter correction, the dataset 90m-nadir gives the lowest accuracy among the three block configuration datasets. By applying the correction, the dataset 30-nadir sees the least improvement and gives the lowest accuracy.

Table 15: similar to *AgiSoft MetaShape*, the rolling shutter correction does not work as expected for corridor configuration dataset. The unfavorable flight configuration makes it difficult to estimate 14 camera calibration parameters per image.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	11.1	10.1	4.6	1.9	-0.5	1.8	11.2	10.2	4.7	c-8p-GCP1	0.8	0.8	0.3	4.8	-0.4	4.7	4.8	3.7	3.1	93	-153
o-8p-GCP2	15.7	13.7	7.7	2.3	0.9	2.1	15.8	13.9	7.6	c-8p-GCP2	2.1	1.8	1.0	5.6	2.7	4.9	6.0	5.0	3.2	87	-143

Table 12: *Pix4D* - Results for dataset 1: nadir-30m.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	16.0	13.0	9.4	7.0	-4.8	5.1	17.4	14.6	9.5	c-8p-GCP1	1.1	1.0	0.5	2.6	-0.1	2.6	2.8	2.5	1.3	93	63
o-8p-GCP2	16.3	13.8	8.6	7.4	3.2	6.7	17.9	15.6	8.8	c-8p-GCP2	1.1	1.1	0.4	1.4	0.3	1.3	1.8	1.6	0.8	93	81

Table 13: *Pix4D* - Results for dataset 2: nadir-90m.

	Original										Corrected									Improvement
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)			on 3D (%)
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	14.6	13.1	6.5	2.4	0.9	2.3	14.8	13.3	6.4	c-8p-GCP1	1.5	1.3	0.7	1.2	0.5	1.1	1.9	1.7	0.8	90	50
o-8p-GCP2	13.4	12.1	5.9	3.3	-2.1	2.6	13.8	12.6	5.6	c-8p-GCP2	1.5	1.4	0.7	1.7	-0.8	1.5	2.3	1.8	1.4	89	49

Table 14: *Pix4D* - Results for dataset 3: 30m-nadir + 60m-oblique + 90m-nadir.

	Original										Corrected									Improvement on 3D (%)	
	Planimetry (cm)			Altimetry (cm)			3D (cm)				Planimetry (cm)			Altimetry (cm)			3D (cm)				
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD		RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	Planimetry	Altimetry
o-8p-GCP1	5.9	3.7	4.6	22.2	11.1	19.2	22.9	14.3	18.0	c-8p-GCP1	62.6	37.7	50.0	38.1	10.8	36.5	73.3	52.0	51.6	-961	-72
o-8p-GCP2	5.9	4.1	4.3	2.7	1.7	2.1	6.5	3.2	5.7	c-8p-GCP2	11.9	9.7	7.0	33.4	7.5	32.5	35.4	8.3	34.4	-102	-1137

Table 15: *Pix4D* - Results for dataset 4: 40m-nadir.

6.4.4. Comparison of results

For each dataset, the results issued from the three software are assembled for comparison purpose and are presented in Figure 13 - 16.

Figure 13: For all three software, the rolling shutter correction is able to improve the obtained accuracy when processed with the 8-parameter camera calibration model. The three software give comparable results with *Pix4D* being slightly on top. When employing the 10-parameter camera model, the rolling shutter correction of *MetaShape* worsens the results, which is probably due to over-parameterization. On the contrary, the proposed method implemented in *MicMac* still works in this case and gives similar results to cases with the 8-parameter model.

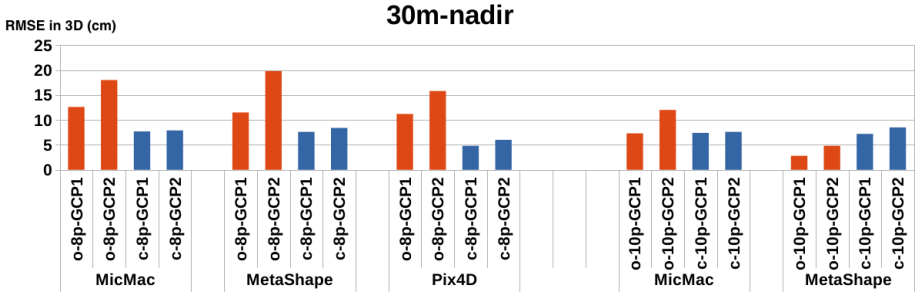


Figure 13: Result comparison of dataset 1: 30m-nadir

Figure 14: A similar trend to Figure 13 is observed. What is worth noting is that, unlike the dataset 30m-nadir, for 90m-nadir, *MetaShape* seems to encounter difficulties with the configuration c-8p-GCP2 and is not able to reach the same accuracy as other two software.

Figure 15: With a good flight configuration being presented, all three software are able to correct the rolling shutter effect as expected. Since the proposed is implemented based on a coarse velocity estimation (the time related informa-

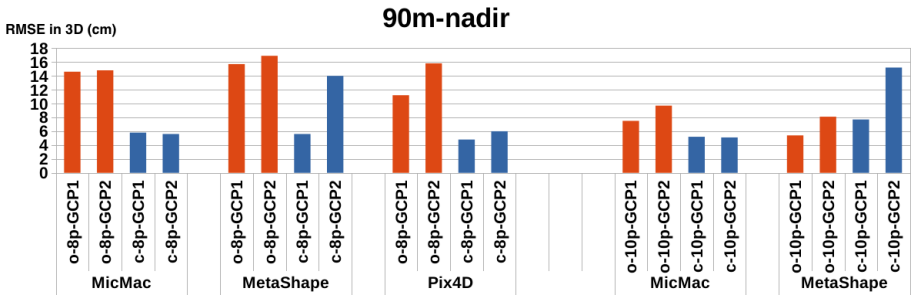


Figure 14: Result comparison of dataset 2: 90m-nadir

tion is rounded to second), it shows less satisfying results for corrected datasets compared to the other two software.

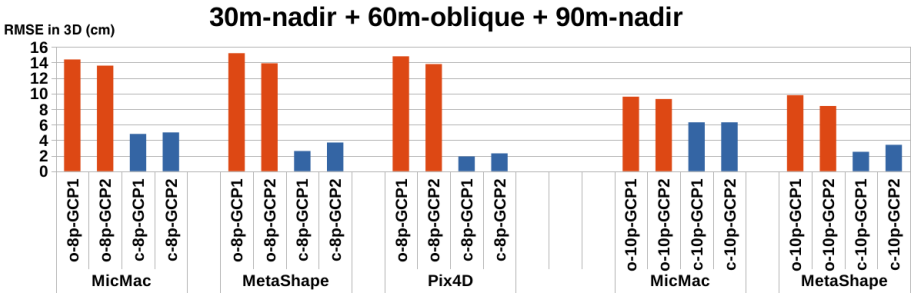


Figure 15: Result comparison of dataset 3: 30m-nadir + 60m-oblique + 90m-nadir

Figure 16: When it comes to the unfavorable corridor configuration, only the proposed method implemented in *MicMac* is able to correct the rolling shutter effect and to improve the accuracy. The other two software all give degraded accuracy after applying the correction. The poor flight configuration makes it more difficult to estimate extra parameters in bundle block adjustment.

7. Conclusion and perspective

In this paper, the rolling shutter effect in aerial photogrammetry is addressed. We proposed an approach to calibrate the readout time of rolling shutter cameras as well as a two-step method for the rolling shutter effect cor-

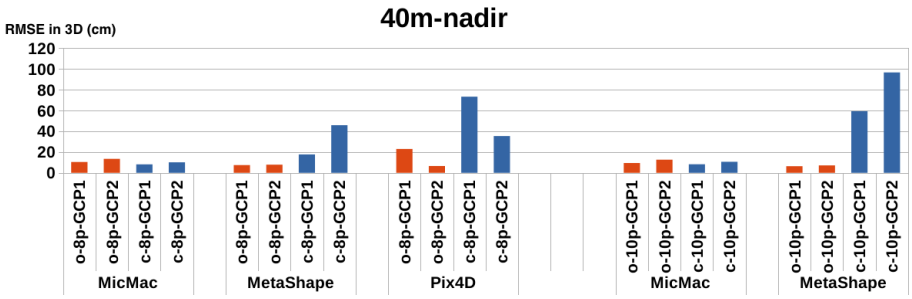


Figure 16: Result comparison of dataset 4: 40m-nadir

rection. The approach of readout time calibration gives the possibility to have control on this information which is often not given by the camera manufacturer. It as well allows the correction of image deformation introduced by this effect. The rolling shutter effect correction is implemented in *MicMac*, a free, open-source photogrammetric software.

The proposed rolling shutter correction method performs under the assumption that the deformation due to camera orientation change is negligible compared to that of camera position change. In our case, the datasets are acquired with a camera which is alleged to provide a stable, smooth footage for image acquisition thanks to a 3-axis gimbal, which conforms this assumption. However, the time related information registered in the image metadata is rounded to second, which can limit the performance of the presented method. Despite this, with a consumer-grade drone/camera and the provided information in metadata, the approach presented in this article is able to improve the accuracy of photogrammetric results with a robust performance.

Four datasets are processed, two nadir-looking block configuration datasets at a flight height of 30m and 90m, respectively; one block configuration dataset consisting of the two first datasets and one oblique-looking dataset at 60m; and one corridor configuration dataset conducted at 40m. Each dataset is processed with two camera calibration models: a 8-parameter model and a 10-parameter model which has two affine distortion coefficients in addition. The ground points are separated into two sets to serve as GCPs and CPs; each dataset is processed

twice with the 2 point sets as GCPs to eliminate the occasionality. The proposed method improves the accuracy by 30%-60% for block configuration and 15%-25% for corridor configuration. Comparison is conducted with other two commercial software, *AgiSoft MetaShape* and *Pix4D*.

Several conclusions can be drawn. To begin with, the inclusion of affine distortion coefficients improves the results when the rolling shutter effect is present. It is comprehensive since the deformation introduced by rolling shutter effect is similar to an affine distortion on a flat scene. *Pix4D* only supports the 8-parameter camera calibration model, no further studies can be carried out. For *AgiSoft MetaShape*, when applying a rolling shutter effect correction, it is recommended not to include the affine distortion coefficients since there is a risk of over-parameterization. The proposed method works with both camera models without the risk of over-parameterization and gives comparable results to other two software.

Secondly, the flight configuration is an essential factor for rolling shutter correction in *AgiSoft MetaShape* and *Pix4D*. For each image, several parameters in addition are estimated during the bundle block adjustment to describe the change of camera pose during acquisition. When a favorable block configuration is presented, such as for instance, nadir images of different flight heights as well as oblique images, both software turn out to work fine. For a less favorable configuration where there is only nadir images with one single flight height, the performance of the rolling shutter correction degrades. In certain cases, *AgiSoft MetaShape* even shows reduced accuracy after applying the correction. For corridor configuration, none of these two software works as expected. On the contrary, the method proposed in this paper is robust to the flight configuration. To correct the deformation in image space, the estimation of the camera velocity during exposure is carried out after the bundle block adjustment, therefore does not have specific requirements on the flight configuration. It is the only method that works for both block and corridor configuration datasets.

Though the method proposed in this article gives satisfying results in all configurations, in the case where a good flight configuration is present (oblique

and nadir images of different flight heights), the other two methods which estimate the camera pose variation show better performance. It is partly due to the compromised time related information explored for camera velocity estimation in this two-step method. A further improvement is expected if an accurate timestamp is provided in the metadata or if the camera positions are directly available with the aid of, for example, GNSS data. Moreover, the assumption of the camera orientation variation being negligible compared to the camera position change can slightly degrade the performance of the proposed method. Future studies should be carried out on the exploration of gimbal data, and how it can be used to estimate the camera orientation change and to perform a complete correction of rolling shutter effect. In this article, experiments were conducted with a high quality camera. The work should be extend to other image sensors to see if same results can be achieved.

References

References

Ait-Aider, O., Andreff, N., Lavest, J.M., Martinet, P., 2006. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera, in: European Conference on Computer Vision, Springer. pp. 56–68.

Cheng, Y.Q., Wang, X., Collins, R.T., Riseman, E.M., Hanson, A.R., 2001. Three-dimensional reconstruction of points and lines with unknown correspondence across images. International Journal of Computer Vision 45, 129–156.

Chun, J.B., Jung, H., Kyung, C.M., 2008. Suppressing rolling-shutter distortion of cmos image sensors by motion vector detection. IEEE Transactions on Consumer Electronics 54, 1479–1487.

Fraser, C.S., 1997. Digital camera self-calibration. ISPRS Journal of Photogrammetry and Remote sensing 52, 149–159.

Grundmann, M., Kwatra, V., Castro, D., Essa, I., 2012. Calibration-free rolling shutter removal, in: 2012 IEEE international conference on computational photography (ICCP), IEEE. pp. 1–8.

Hedborg, J., Ringaby, E., Forssén, P.E., Felsberg, M., 2011. Structure and motion estimation from rolling shutter video, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE. pp. 17–23.

Ito, E., Okatani, T., 2017. Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 801–809.

Klein, G., Murray, D., 2009. Parallel tracking and mapping on a camera phone, in: 2009 8th IEEE International Symposium on Mixed and Augmented Reality, IEEE. pp. 83–86.

Liang, C.K., Chang, L.W., Chen, H.H., 2008. Analysis and compensation of rolling shutter effect. IEEE Transactions on Image Processing 17, 1323–1330.

Magerand, L., Bartoli, A., 2010. A generic rolling shutter camera model and its application to dynamic pose estimation, in: International symposium on 3D data processing, visualization and transmission.

Martinez-Rubi, O., 2016. Improving open-source photogrammetric workflows for processing big datasets. Netherlands eScience Center: Amsterdam, The Netherlands .

Meingast, M., Geyer, C., Sastry, S., 2005. Geometric models of rolling-shutter cameras. arXiv preprint cs/0503076 .

Nex, F., Remondino, F., 2014. UAV for 3D mapping applications: a review. Applied geomatics 6, 1–15.

Rupnik, E., Daakir, M., Pierrot-Deseilligny, M., 2017. MicMac–a free, open-source solution for photogrammetry. Open Geospatial Data, Software and Standards 2, 14.

Saurer, O., Pollefeys, M., Hee Lee, G., 2016. Sparse to dense 3D reconstruction from rolling shutter images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3337–3345.

Sun, Y., Liu, G., 2012. Rolling shutter distortion removal based on curve interpolation. IEEE Transactions on Consumer Electronics 58, 1045–1050.

Sun, Y., Liu, G., Sun, Y., 2016. An affine motion model for removing rolling shutter distortions. IEEE Signal Processing Letters 23, 1250–1254.

Vautherin, J., Rutishauser, S., Schneider-Zapp, K., Choi, H.F., Chovancova, V., Glass, A., Strecha, C., 2016. Photogrammetric accuracy and modeling of rolling shutter cameras. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci 3, 139–146.

Zhou, Y., Rupnik, E., Meynard, C., Thom, C., Pierrot-Deseilligny, M., 2019. Simulation and analysis of photogrammetric UAV image blocks: influence of camera calibration error. Submitted to Remote Sensing, currently under revision .