A SYSTEM FOR MONITORING OF UAV CAMERA ORIENTATION: DESIGN AND INITIAL ANALYSIS

J. Tekavec [1, *], K. Oštir [1], A. Lisec [1], G. Štebe [1]

[1] University of Ljubljana, Faculty of Civil and Geodetic Engineering, Ljubljana, Slovenia - (jernej.tekavec, kristof.ostir, anka.lisec, gasper.stebe)@fgg.uni-lj.si

KEY WORDS: UAV, rolling shutter, vibration, MEMS, gyroscope

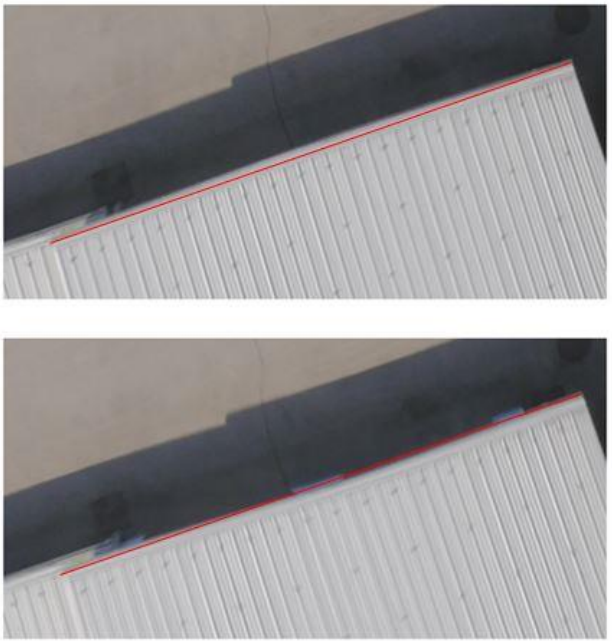
ABSTRACT:

Many unmanned aerial vehicles (UAV) that are used for aerial mapping are equipped with consumer-grade digital cameras, which use CMOS (Complementary metal–oxide–semiconductor) image sensors. Majority of these sensors capture images using an electronic rolling shutter, which can cause distortions on the image if either the camera or the captured objects are moving. This phenomenon is usually ignored in aerial mapping with UAVs in practice. However, there is a lack of published research papers that would prove the effect can be neglected. In this paper, we present the design of the system for monitoring UAV camera orientation. Furthermore, the calibration process to get correct and reliable readings is described. The initial analysis of the data is focused on assessing the accuracy that can be achieved using the proposed system. The main component of our system is a MEMS (Microelectromechanical system) gyroscope. It was selected for its low weight and size, low price and high sampling rates which are all very beneficial characteristics for a system, mounted on a UAV. In a paper, a working prototype is presented that uses the selected MEMS gyroscope connected to a single-board computer. The presented initial analysis of collected data shows, that the system would be capable to indirectly detect the image distortions, caused by camera orientation changes during exposure, in the range of typical ground sample distance (GSD).

1. INTRODUCTION

UAVs are becoming a widely used tool for spatial data acquisition. They are both replacing and supplementing traditional surveying techniques in many applications, as they shorten the acquisition time, reduce costs and even improve the accuracy of some products, like volumes of excavated materials. The UAVs gained popularity with the first affordable autopilots approximately ten years ago, allowing user-friendly piloting, good stability, and navigation along a pre-planned route. The UAVs are nowadays technically advanced and affordable platforms for spatial data acquisition. The software for photogrammetric processing of UAV imagery experienced similar evolution regarding accuracy, processing speed and computing load, point cloud processing, and feature extraction. The software combined with UAV as a platform, and a camera, provide a comprehensive package for efficient spatial data acquisition. Despite all this progress, a very important component of this package from the photogrammetric point of view was not considered – until recently, there has been no evident development of small and light metric cameras. Nowadays, a big majority of UAVs are equipped with consumer-grade cameras that use CMOS image sensor technology. CMOS image sensors are better than CCD (charge-coupled device) image sensors in many aspects that are important for mass marketing, but not in the one, that is very important in photogrammetry – constant and stable image geometry. CMOS image sensors use an electronic rolling shutter that causes image distortions in case either the camera or the object is moving. Because UAV is far from being a stable platform in the air, research is needed to develop systems and methodologies for estimation, reduction, and correction of the errors, caused by the instability.

The main motivation for our work was the amount of CMOS image sensor-based cameras usage on UAVs. The mentioned instability of the camera with a CMOS image sensor, mounted on the UAV causes the image distortions that can be clearly seen in Figure 1. The images were acquired with UAV on a windy day with poor CMOS image sensor camera stabilization and demonstrate, the extent of image distortions.



* Corresponding author

Figure 1. A subset from two sequent images taken from UAV with CMOS camera in windy conditions (our archive)

2. REVIEW OF PAST RESEARCH ACTIVITIES

Most of the research related to the investigation of rolling shutter effect has been done in the field of machine vision. Gryer et al. (2005) and Ringaby (2014) presented and described the mathematical background of rolling shutter image acquisition and its effect on the image.

Ait-Aider et al. (2006) showed that rolling shutter effect in the image could be used for defining the object motion parameters. This, in turn, means that if we want to remove the distortions, caused by the rolling shutter effect, we must know how both the camera and all objects in the picture were moving when the image was taken.

Various methods considering the rolling-shutter effect were proposed in Nicklin et al. (2007), Chun et al. (2008), Liang et al. (2008), Baker et al. (2010), Hedborg et al. (2011), Grundmann et al. (2014). Baker et al. (2010) and O'Sullivan et al. (2014) emphasized the importance of calibrating the time delay between each image row is captured. Most proposed methods are not focused on correcting a single image, but use images from videos, which is basically the time series of images. Karpenko et al. (2011) and Jia et al. (2012) used motion sensors for camera movement estimation, as this kind of sensors are becoming widely used in modern electronic devices. The focus of their and research is mitigation of so-called "jello effect" on the captured video.

Recently Vautherin et al. (2016) published an article that focuses on the photogrammetric assessment of rolling shutter cameras in case they are used on UAV. They present the solution, where rolling shutter effect on the captured image is corrected for camera translations during exposure. The authors presented how rolling shutter affects not only images but also the results of the photogrammetric processing of these images. They focused on modeling and correcting camera translational movement. The results are significantly improved when they applied the proposed algorithm. The improvement of the result is correlated to UAV's speed, as the rolling shutter effect increases with the speed. This proves the rolling shutter effect is significant in aerial mapping applications. The proposed solution is more appropriate for fixed wing UAVs. Multirotor UAVs, on the other hand, fly at lower speed but are much more prone to vibrations that cause fast camera orientation changes.

3. RESEARCH METHODOLOGY

As the effect of the rolling shutter image sensor rotation on the image is well known, the main contribution of this paper is the design of a system, capable of monitoring orientation changes during the exposure time. The aim of the research is not the correction of the image distortions, but to get the information about camera orientation changes during UAV flight. Using this information, it is possible to assess the magnitude of distortions that can occur on the captured images.

First, the effects of different types of CMOS image sensor movement were studied and analysed. Figure 2 shows the effects of image sensor translation and rotation. To correct these distortions, the camera movement, object movement and also object distance from the camera should all be known, which makes it very challenging.

Figure 2. Image captured from moving car (up) and by rotating the CMOS image sensor (bottom) (our archive)

The key component of our system is a MEMS gyroscope. The sensor was first calibrated and then tested for noise and accuracy of its readings to check if it can deliver useful data for our application. Based on the selected gyroscope, other components were selected for the collection and storage of the sensor data, power supply and mounting on the UAV.

4. SYSTEM DESIGN

In photogrammetry, the typical analysis workflow is first taking pictures and then analysing and correcting them for various effects (optical aberrations etc.). We decided to use a different approach and try to measure the cause of the distortions at its source with the system, mounted on the UAV during its operation.

The design of a system for monitoring of UAV camera orientation consisted of the following steps:

- sensor selection,
- testing sensor characteristics,
- assembling and mounting the system on the UAV,
- collecting and analysing the data from real UAV flight.

4.1 Gyroscope sensor selection

The initial parameters we considered in gyroscope sensor selection were sensor cost, availability, size and weight, sensitivity, sampling rate, and the declared accuracy. First, we limited the selection to MEMS gyroscopes as they are highly affordable. The size and weight enabled mounting directly on the camera without significantly changing its momentum characteristics i.e. not influencing the camera movement.

The biggest disadvantage of all affordable MEMS gyroscopes is that they have large angular drift over time, which limits or prevents their use in many applications. However, in our case, we largely avoided this problem, as the time interval, which we are interested in, is very short. The readout time of CMOS image sensors, used in cameras on UAVs is generally around 30 milliseconds in most cases (Ringaby, 2014, O'Sullivan et al., 2014, Vautherin et al., 2016). More advanced CMOS imaging sensors have readout times of 10 milliseconds (LaBelle, 2014) and less. However, the article addresses the consumer grade CMOS sensor cameras, that are nowadays being predominantly used in practice for UAV mapping purposes.

We started with MPU 6050 sensor from InvenSenese that combines 3-axial accelerometer and 3-axial gyroscope. The declared maximum sample rate for the gyroscope is 8 kHz. Connected to Arduino microcontroller using I2C bus and i2cdevlib libraries (Rowberg, 2018) the maximum effective sampling rate we could achieve with logging the results was approximately 200 Hz, which is relatively low for given readout time of 30 milliseconds. The second choice for further research was the InvenSense MPU 9250, which supports faster SPI bus (compared to I2C bus) communication and can sample at the same sample rate of 8 kHz. After testing MPU 9250, connected over SPI bus, it became clear that the bottleneck is the speed of data logging. Our solution was reading data from the sensor into a biggest possible vector and when it is full, log the whole vector. Thus, the data logging is not decelerating the sampling rate but is interrupting the sampling at regular intervals. This could not be realized using Arduino microcontroller for its very limited space available for declared variables. Instead, we used Raspberry Pi 2 Model B and C++ libraries (Avkhimenia, 2018). This modification allowed us to allocate three vectors, each for one of the three axis readings that can store 500 000 variables. The sensor connection to Raspberry Pi is done using the SPI bus. The achieved sampling speed was between 2 and 4 kHz. This means we could get 60–120 samples per one image sensor readout time.

4.2 Testing gyroscope characteristics

Raw sensor readings represent scaled angular velocity. Their scale depends on the selected measurement range (full-scale range). To get the sensor orientation, the raw readings need to be scaled, corrected for offset and integrated over time. The integration of angular velocity causes the already mentioned drift of sensor orientation values.

The scale of the raw readings can be obtained in sensor factory datasheet for each range. We selected the ± 250°/sec measuring range, with corresponding, factory determined scale factor of 131. The offset for each axis is different and also changes over time. It is therefore recommended to perform offset calculation right before measurement. To do this, we implemented a basic correction algorithm, which requires the sensor to be as still as possible when executing. It collects raw readings for arbitrarily selected time of 5 seconds, calculates the mean value for each axis and stores it as the axis offset. More advanced algorithms for gyroscope bias determination and temperature compensations can be found in Aggarwal (2008), Fang (2013) and Anderson et al. (2015).

4.2.1 Testing the sensor noise: The noise can be characterized as a random scattering of the readings. Within the noise range, no useful information can be extracted from the readings. The time interval we are interested in for noise calculation is the image sensor readout time, ranging from 10–50 milliseconds for most CMOS image sensors. Given our sampling rate of 2–4 kHz, we determined the base frame of 120 sequent values to calculate the standard deviation and value range. We mounted the sensor on a fixed and stable base and collected a sample of 40,000 readings for each axis. The raw readings were corrected for scale and offset and integrated over time to get the gyroscope sensor orientations for each axis. The frame of 120 values was moved through all values with one value step. For each step, the standard deviation **σ**, and value range **r** were calculated based on selected 120 values (Figure 3).

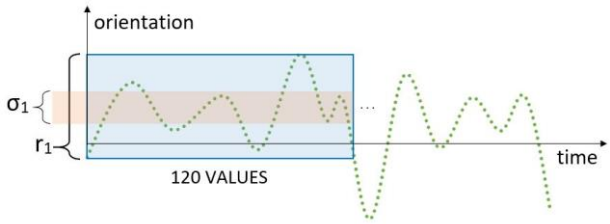


Figure 3. **σ** and **r** for the first "frame" of 120 values

Thus, we got 39,880 **σ** and **r** values based on 40,000 original values. Below in Table 1, the mean values for **σ** and **r** are presented for each axis and 3 separate tests. Table 2 contains the maximum **σ** and **r** among all 39,880 values, again for each axis.

Axis	X		Y		Z	
	σ ["]	**r** ["]	**σ** ["]	**r** ["]	**σ** ["]	**r** ["]
Test 1	1.1	3.8	1.3	4.7	1.1	3.8
Test 2	1.0	3.6	1.2	4.5	1.1	3.8
Test 3	1.1	3.9	1.3	4.5	1.1	3.8

Table 1. The mean of **σ** and **r** for 3 separate tests

Axis	X		Y		Z	
	σ ["]	**r** ["]	**σ** ["]	**r** ["]	**σ** ["]	**r** ["]
Test 1	2.7	10.0	4.4	13.4	3.0	9.0
Test 2	3.0	9.5	3.7	12.1	2.8	9.4
Test 3	3.5	10.3	3.6	11.6	3.4	11.0

Table 2. The maximum of **σ** and **r** for 3 separate tests

From the tables above, we can see that all value ranges **r** are smaller than 15", with most of them being smaller than 10". If we put those values in the scope of UAV, knowing its altitude above ground, we can calculate what this noise means for detecting the image distortions, caused by the rotational movement of the image sensor. Based on the presented tests, we can assume, that we are able to detect rotations larger than 15" within the time frame of 30-60 milliseconds. Assuming the altitude of UAV is 50 m, the distortions, caused by 20" rotations during image sensor readout, result in 5 mm positional error on the ground (Figure 4).

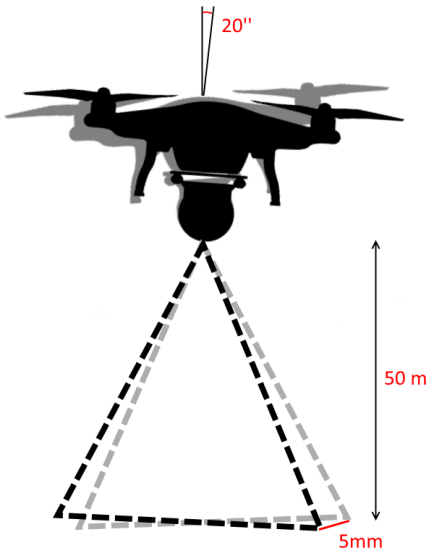


Figure 4. Image distortions caused by 20" image sensor rotation during sensor readout at 50 m altitude (svgsilh.com, 2019)

4.2.2 Testing the sensor accuracy: In the second test, we assessed the accuracy of sensor orientation, calculated from sensor readings. To assess the accuracy, the "true" value should be known or measured. We fixed the sensor to the Leica TS30 total station telescope, which features 0.5" angular accuracy, enough to take its measurements for the reference (Figure 5).

Figure 5. Leica TS 30 with gyroscope installed on the telescope

Connected to Raspberry Pi, the instrument can accurately rotate for given angles without any contact needed. When performing the tests, the total station base was fixed and still. After collecting data for initial offset calculation, we sent commands for several 1° rotations to the total station (Figure 6). The time needed for instrument rotation was between 0.2 and 0.3 seconds.

It is practically impossible to mechanically align the sensor axes with total station axes. This means the sensor orientation change cannot be calculated directly from one axes readings. As can also be seen from in Figure 6, the total station rotation around one axis is reflected in rotation of all three gyroscope sensor axes.

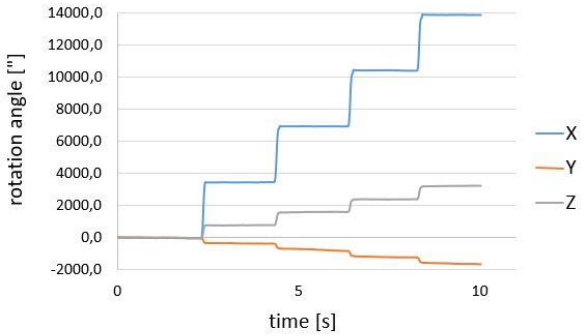


Figure 6. Leica TS 30 telescope orientations, calculated from gyroscope sensor readings

By calculating the rotation matrix, the telescope rotation angle θ can be calculated from the equation for matrix trace calculation.

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \quad (1)$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad (2)$$

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$R = R_z(\alpha) * R_y(\beta) * R_x(\gamma) \quad (4)$$

$$trace(R) = 1 + 2 * \cos\theta \quad (5)$$

To reduce the influence of orientation drift and sensor noise, we manually selected 100 orientation values right before and after each rotation and used the mean of selected values. Table 3 contains these means for the start and the finish of each rotation. The difference is then calculated using above equations and compared to the reference angle of 1° measured by total station.

	Telescope orientation X ["]	Telescope orientation Y ["]	Telescope orientation Z ["]	Difference to 1° ["]
Before After	-51 3433	-41 -340	-23 757	18
Before After	3448 6927	-384 -691	772 1563	20
Before After	6927 10421	-841 -1172	1590 2375	4
Before After	10393 13885	-1240 -1573	2385 3186	3

Table 3. Averaged telescope orientations and the difference between calculated θ and 1° for each rotation

Knowing that we got orientation values with integration over time, any significant remaining scale error would be discovered in this test. Based on the tests we can conclude that the MPU 9250 MEMS gyroscope sensor is suitable for monitoring the image sensor orientation during UAV flight. The next step is to assemble the system that can be mounted to UAV.

4.3 Assembling and mounting the system on the UAV

The components of the system that need be mounted on the UAV are the Raspberry Pi computer, MPU 9250 sensor, and power supply. The UAV power supply can be used, but to achieve the system independence and avoid electrical disturbances, the additional power supply was chosen in our case. The Raspberry Pi placement can vary depending on the UAV structure layout. The cable for SPI connection runs from Raspberry Pi to the gyroscope that should be mounted to the gimbal or directly to the camera. This assures that the sensor is exposed to the same conditions as the image sensor in the camera (Figure 7).

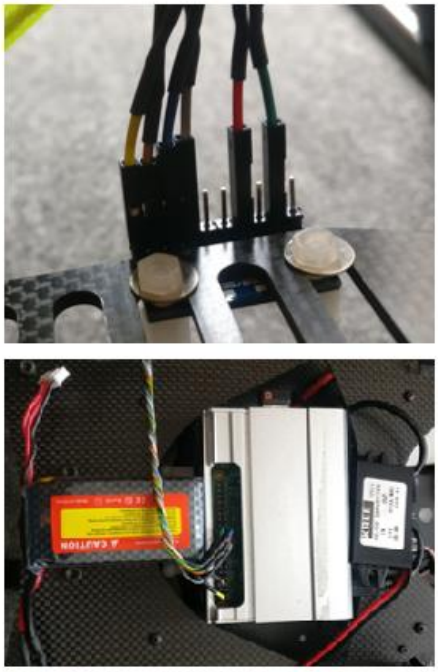


Figure 7. A gyroscope mounted on the gimbal (up) and Raspberry Pi with the power supply on the upper side of the Sky Hero 850 frame (bottom)

4.4 Collecting UAV flight data and initial analysis

We selected a medium heavy UAV for the test purposes, which has enough lifting power for additional components. It is based on SkyHero 850 X8 frame. For lighter and less powerful UAVs, the system's weight and size should be optimised.

We collected 500.000 gyroscope sensor readings during UAV flight, which results in 151 seconds of flight time. The sensor readings for initial offsets calculation were collected right before the flight with UAV motors not spinning to reduce vibrations. The temperature during the flight was around 20°C with windspeed not exceeding 5 km/h.

After the flight, the readings were transferred from Raspberry Pi to the PC for the analysis. The initial offsets were calculated for all three gyroscope axes as the average value of the readings collected for 5 seconds. These offsets were deducted from original sensor readings. Time delay between each subsequent pair of readings was calculated from the timestamps collected for each sensor reading. Using time delay and sensor readings corrected for the offsets (bias), the sensor orientations were calculated for each axis.

We analysed the orientation changes from the perspective of the distortion that they cause, measured on the ground, on images taken from 50 m altitude from the ground (Figure 4). The aim of this analysis is to get the percentage of the (analysed) flight time, when the camera orientation changes cause image distortions greater than the selected value. We selected two values for the analysis, 5 cm and 10 cm respectively. The calculation of orientation changes requires selection of the time frame within which the changes are calculated (similar as for sensor noise estimation), which has to correspond to the COMS image sensor readout time. We selected 10 ms and 20 ms time frame for the analysis. With the achieved gyroscope sampling rate of 4 kHz this translates to the selection of 40 and 80 subsequent sensor readings.

For each "window" of 40 and 80 subsequent readings, the value range was calculated. It represents the maximum orientation change of the camera within the selected time frame. At 50 m altitude, the orientation change, needed for 5 cm ground measured distortion of the image, is approx. 205". For 10 cm distortion, the orientation change should be approx. 410" (see Figure 4). We added 20" to both threshold values, to account for the sensor noise, which has been tested to be lower than 20" during the presented research. If orientation change for any of the axis was exceeding the threshold value, the time frame was flagged. The table 4 summarizes the results by showing the percentage of the flagged time frames.

	Time frame	
Distortion	10 ms	20 ms
5 cm	25%	56%
10 cm	11%	25%

Table 4. The percentage of the flight when the orientation changes exceeded the threshold value for at least one axis

5. DISCUSSION

The system offers various implementations, of which the basic one was selected for initial tests. Its aim is to give an overview of the camera short-term orientation changes during the flight. Knowing the approx. readout time of the used CMOS image sensor one can calculate the percentage of the flight time, when the orientation changes exceeded the threshold values. While the latter does not offer any possibility to improve the images nor the results, it offers an estimate of conditions in which the camera was capturing images.

The more advanced implementation could use logging of the camera trigger signal from autopilot, to get an estimate of when the image was taken. To synchronise with gyroscope, the trigger signal should be logged using the same system clock as used for logging the gyroscope readings. The calculation of orientation changes can be narrowed down to short time intervals when each of the images was captured. Using this information, it is possible to exclude the images captured in the conditions that cause the distortions that exceed the selected limit. To further narrow down the time interval if image capture, the trigger signal from the camera can be used, normallya used for flash synchronisation.

The third possible implementation could provide information to correct the images for the errors caused by orientation changes. This task is very challenging from many aspects. The time interval of the image readout would have to be very precisely determined together with precisely logging its start. The frequency of gyroscope sampling and logging would have to be increased. The gyroscope sensor errors would have to be determined more accurately (Stebler et al., 2014) and rigorously tested. Despite doing everything listed above only a part of rolling shutter effect, caused by orientation changes, could theoretically be removed. To meet all presented requirements, the orientation monitoring system would have to be tightly integrated with the camera, not being a separate stand-alone system, which is out of the scope of this article.

The proposed system could leverage the two available SPI bus connections on Raspberry Pi to connect two separate gyroscope sensors. This opens possibilities to improve the quality of measurements by fusing the data from both sensors.

Initial analysis of the data, collected during UAV flight suggests that the angular velocity of the camera, causing changes in orientation, exceeded the distortion limit in a considerable part of the flight. To simplify the analysis, the calculations were performed separately for each axis instead of using equations (1) – (5) to calculate the rotation angle θ and checking if θ exceeds the threshold value. By checking each axis separately, we stayed at the safe side as θ is always larger than its components α, β and γ.

6. CONCLUSIONS

The article proposes a system for monitoring UAV camera orientation during the flight comprised of affordable components. The focus is on the methodology of the system design together with noise and accuracy tests that confirm its applicability. Both static sensor noise and short-term rotation determination accuracy are below 20", which corresponds to approximately 5 mm distortion error at 50 m flight altitude. Very short readout times of CMOS image sensors make the most problematic limitation of MEMS gyroscope – its drift – practically insignificant.

The system can be applied in the UAV design phase, to determine, if the used combination of UAV and gimbal can provide a stable platform for the camera. Separate analysis of gyroscope readings for each axis can identify which axis has to be additionally stabilised.

The best approach in UAV mapping is to completely avoid the rolling shutter effect by using global shutter image sensors. Also, some newer and more advanced CMOS sensors have shorter readout times than presented in the article, thus reducing the image distortions. However, the article's focus is on the consumer grade CMOS cameras, nowadays predominantly used on UAVs for mapping purposes. The added value of the proposed system is not to enable image correction, but to provide information if the camera was stable enough during the flight and there are no significant distortions on the captured images.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0406 Earth observation and geoinformatics).

REFERENCES

Aggarwal, P., Syed, Z., El-Sheimy, N., 2008. Thermal calibration of low cost MEMS sensors for land vehicle navigation system. *IEEE Vehicular Technology Conference*, 2859–2863. doi.org/10.1109/VETECS.2008.623.

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 Leonardis, A., Bischof, H., Pinz, A. (Eds.), *Computer Vision - ECCV 2006, Lecture Notes in Computer Science*, 56–68. doi.org/10.1007/11744047_5.

Anderson, S., MacTavish, K., D. Barfoot, T. 2015. Relative continuous-time SLAM. *The International Journal of Robotics Research*, 34.12: 1453-1479. doi.org/10.1177/0278364915589642

Avkhimenia, M., 2018. Emlid/Navio2. GitHub repository. https://github.com/emlid/Navio2 (15 December 2018).

Baker, S., Bennett, E., Kang, S. B., Szeliski, R., 2010. Removing rolling shutter wobble. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2392–2399. doi.org/10.1109/CVPR.2010.5539932.

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(4), 1479–1487. doi.org/10.1109/TCE.2008.4711190.

Fang, B., Chou, W., Ding, L., 2013. An Optimal Calibration Method for a MEMS Inertial Measurement Unit. *International Journal of Advanced Robotic Systems*, 11(2), 4. doi.org/10.5772/57516.

Geyer, C., Meingast, M., Sastry, S., 2005. Geometric models of rolling-shutter cameras. *Proceedings of Omnidirectional Vision, Camera Networks and Non-classical Cameras*.

Grundmann, M., Kwatra, V., Castro, D., Essa, I., 2012. Calibration-free rolling shutter removal. *IEEE International Conference on Computational Photography*, 1–8. doi.org/10.1109/ICCPhot.2012.6215213.

Hedborg, J., Ringaby, E., Forssen, P.-E., Felsberg, M., 2011. Structure and motion estimation from rolling shutter video. *IEEE International Conference on Computer Vision*, 17–23. doi.org/10.1109/ICCVW.2011.6130217.

Jia, C., Evans, B. L., 2012. Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements. *IEEE 14th International Workshop on Multimedia Signal Processing*, 203–208. doi.org/10.1109/MMSP.2012.6343441.

Karpenko, A., Jacobs, D., Jongmin, B., Levoy, M., 2011. *Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes*. Stanford University Computer Science Tech Report CSTR 2011-03. https://graphics.stanford.edu/papers/stabilization/ (14 January 2019).

Labelle, R., 2014. Global-Shutter vs. Rolling-Shutter Readouts. BioPhotonics, Dec 2014. https://www.photonics.com/Articles/Global-Shutter_vs_Rolling-Shutter_Readouts/p1/vo118/i792/a57010 (14 January 2019).

Liang, C., Member, S., Chang, L., Chen, H. H., 2008. Analysis and Compensation of Rolling Shutter Effect. *IEEE Transactions on Image Processing*, 17(8), 1323–1330. doi.org/10.1109/TIP.2008.925384.

Nicklin, S. P., Fisher, R. D., Middleton, R. H., 2007. Rolling Shutter Image Compensation. In Lakemeyer G., Sklar E., Sorrenti D. G., Takahashi T. (Eds*). RoboCup 2006: Robot Soccer World Cup X. Lecture Notes in Computer Science*, 4434. Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-540-74024-7_39.

O'Sullivan, L., Corke, P., 2014. Empirical modelling of rolling shutter effect. *IEEE International Conference on Robotics and Automation*, 2132–2137. doi.org/10.1109/ICRA.2014.6907152.

Ringaby, E., 2014. *Geometric Models for Rolling-shutter and Push-broom Sensors*. Doctoral thesis. Linköping University, Sweden. 180p.

Rowberg, J., 2018. i2cdevlib. GitHub repository. https://github.com/jrowberg/i2cdevlib (5 December 2018).

Stebler, Y, Guerrier, S., Skaloud, J., Victoria-Feser, M.-P., 2014. Generalized method of wavelet moments for inertial navigation filter design. *IEEE Transactions on Aerospace and Electronic Systems* 50.3 (2014): 2269-2283.

Svgsilh.com, 2019. SVG > drone camera. https://svgsilh.com/image/1750345.html (14. January 2019).

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 Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, 139–146. doi.org/10.5194/isprsannals-III-3-139-2016.