

Article

Not peer-reviewed version

Self-calibration of UAV Thermal Imagery Using Gradient Descent Algorithm

Radosław Szostak , [Mirosław Zimnoch](#) ^{*} , [Przemysław Wachniew](#)

Posted Date: 4 October 2023

doi: 10.20944/preprints202310.0187.v1

Keywords: UAV; thermal images; surface temperature; calibration



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Self-Calibration of UAV Thermal Imagery Using Gradient Descent Algorithm

Radosław Szostak, Mirosław Zimnoch * and Przemysław Wachniew

AGH University of Krakow, Faculty of Physics and Applied Computer Science, Reymonta St. 19, Kraków, Poland

* Correspondence: zimnoch@agh.edu.pl

Abstract: Unmanned aerial vehicles (UAV) thermal imagery offers several advantages in environmental monitoring, as it can provide a low-cost, high-resolution, and flexible solution to measure the temperature of the surface of the land. Limitations related to the maximum load of the drone lead to use of lightweight uncooled thermal cameras whose internal components are not stabilized to a constant temperature. Such cameras suffer from several unwanted effects that contribute to the increase in temperature measurement error from ± 0.5 °C in laboratory conditions, to ± 5 °C in unstable flight conditions. This article describes a post processing procedure, that reduces the above unwanted effects. It consists of following steps: i) devignetting using single image vignette correction algorithm, ii) georeferencing of images using EXIF data, scale-invariant feature transform (SIFT) stitching, and gradient descent optimisation, and iii) temperature calibration by minimisation of bias between overlapping thermal images using gradient descent optimisation. The solution was tested in several case studies of river areas, where natural water bodies were used as a reference temperature benchmark. In all tests, the precision of the measurements was increased. The root of the mean of the Square of Errors RMSE on average was reduced by 39.0% and Mean of the absolute value of Errors MAE by 40.5%. The proposed algorithm can be called self-calibrating, as in contrast to other known solutions is fully automatic, uses only field data and does not require any calibration equipment or additional operator effort. A Python implementation of the solution is available on GitHub.

Keywords: UAV; thermal images; surface temperature; calibration

1. Introduction

1.1. Uncooled thermal cameras issues

Thermal imagery from UAVs has several advantages in environmental applications as they can provide a low-cost, high-resolution, and flexible approach to environmental monitoring and management. UAV-based thermal and narrowband multispectral imaging sensors can provide low-cost approaches to meet the critical requirements of spatial, spectral, and temporal resolutions for vegetation monitoring [1]. UAVs provide high spatial resolution and flexibility in acquisition and sensor integration, which can be used for land cover classification, change detection, and thematic mapping [2]. Aerial thermography from low-cost UAVs can be used to generate digital thermographic digital terrain models, which find application in the classification of land uses according to their thermal response [3]. Thermal remote sensing has many potential uses in precision agriculture, including monitoring plant hydration levels, identifying instances of plant diseases, evaluating crop yield, and analysing plant characteristics [4].

Limitations related to the maximum load of the UAV, as well as cost constraints, lead to the use of mainly lightweight uncooled thermal cameras. This solution has several shortcomings when measuring temperature under field conditions. Uncooled UAV thermal cameras require careful calibration and correction for various factors to obtain accurate temperature measurements. They suffer from vignette effects, sensor drift, ambient temperature influences, and measurement bias, which can be corrected with an ambient temperature-dependent radiometric calibration function [5]. Non-radiometric uncooled thermal cameras are highly sensitive to changes in their internal

temperature and require empirical line calibration to convert camera digital numbers to temperature values [6]. Fluctuations in the temperature of the focal plane array (FPA) array detector, wind, and irradiance can affect temperature measurements, and that adequate settings of camera gain and offset are crucial to obtaining reliable results [7]. Uncooled thermal cameras also suffer from thermal-drift-induced nonuniformity or vignetting [8]. The above factors contribute to fluctuations in camera's accuracy from ± 0.5 °C in laboratory conditions to ± 5 °C in unstable UAV flight conditions [6].

The literature review provides a number of solutions to reduce this problem. To minimise measurement errors in UAV thermal cameras, it is suggested that the camera be warmed up for 15-40 minutes before starting the actual measurement [6,8]. There are also several methods to eliminate unwanted phenomena outcomes with calibration under controlled conditions. Ribeiro-Gomes et al. proposed a calibration algorithm based on neural networks that allows for increased measurement accuracy [9]. In another study by Yuan and Hua, nonuniformity of vignetting was examined in UAV-based uncooled thermal cameras and a simple vignetting reduction method using reference images of a homogeneous target was proposed [8]. In Aragon et al., an ambient temperature-dependent radiometric calibration function was used to correct for sensor drift, ambient temperature influences, measurement bias, and vignette effects [5]. An advanced radiometric calibration approach that stabilises the camera's response, removes fixed pattern noise, and converts thermal image values into object temperatures was proposed by Lin et al. [10]. The disadvantage of calibration-based methods is that they require non-standard equipment and the extra effort needed to collect calibration data. A correction based only on the field data collected by the UAV system (self-calibration) would be favoured. Mesas-Carrascosa et al. developed a method for bias correction using redundant data from overlapping areas of aerial thermal images using mathematical modelling of the metric of "variation of digital number per second" [11]. Unfortunately, despite good results, the authors did not provide a specific explanation, a formula, or sample source code illustrating the calculation of this metric. Also, since this method leverages the rate of digital number variation, it requires precise timing of image acquisition achieved with a custom drone attachment. Often by default the time information available in the image metadata is provided with a precision of a second, which is not sufficient for rate of digital number variation calculation, as during the flight successive photos are taken even as often as every ca. 1 second.

Another important aspect of the problem is the choice of thermal imaging camera brand. Some manufacturers of cameras prevail in publications related to the topic discussed. The files generated by such cameras offer a great deal of freedom to recalibrate temperature measurements, as the sensor response raw data can be easily accessed. Another manufacturer offering in recent years its own thermal imaging cameras dedicated to drones, has another internal processing algorithms policy. In their case, raw sensor data is not openly available, and it is impossible for the user to recalibrate the device. The user has to rely on producer's closed-source Thermal SDK library allowing to make corrections to the measurement taking into account air humidity, target emissivity or flight altitude. It is not known what exact processes are represented in this algorithm. Moreover, we noticed its significant shortcomings: i) the maximum flight altitude acceptable by the library is 25m, which is too low for most flights covering large areas, ii) the library returns errors for some humidity values (in our tests, most often for relative humidity of about 60%), iii) for a given frame it is possible to select only one emissivity (it is not possible to set different emissivities for different objects in the same photo). However, most of the UAV dedicated cameras are susceptible to adverse phenomena typical of uncooled thermal cameras (bias, vignette effect).

1.2. Gradient descent algorithm

A major source of measurement uncertainty is bias, which can be addressed by investigating redundant data from overlapping areas of the images. It is desirable that the temperature difference between all pairs of overlapping images be as small as possible. This is a minimisation problem that can be solved with any optimisation method. This work focusses on the use of gradient descent, which is an optimization algorithm commonly used in machine learning. However, beyond its main

application in machine learning, gradient descent can be used to optimise any differentiable objective function.

Gradient descent works by iteratively updating a set of parameters in the direction of the steepest descent of a cost function. The algorithm computes the gradient of the cost function with respect to the parameters, and then updates the parameters by taking a step in the direction of the negative gradient. The learning rate parameter determines the size of the step and it is usually set to a small value to avoid overshooting the minimum of the cost function. The algorithm continues to iterate until the cost function converges to a minimum or a stopping criterion is met. With the popularity of machine learning, gradient descent has become more accessible thanks to the development of software libraries that accelerate the algorithm by using the GPU (Graphical Processing Unit).

1.3. Objective of this work

The aim of this work is to develop a method for post-processing of aerial thermal imagery that reduces the effects of undesirable phenomena occurring in uncooled thermal cameras without the use of non-standard equipment, such as reference black bodies or custom UAV attachments, and without access to raw thermal sensor data. Achieving this goal is important, as the use of thermal imaging cameras is becoming widespread, but proven post-processing methods are lacking.

2. Materials and Methods

2.1. Data

The thermal pictures were collected by the DJI Matrice 300 RTK drone system equipped with a Zenmuse H20T multicamera sensor that contains the thermal camera. Data used in this study was collected of several areas in southern Poland:

- Area around ca. 500 m Kocinka stream stretch near Grodzisko village (50.8715 N, 18.9661 E)
- Area around ca. 350 m Kocinka stream stretch near Rybna village (50.9371 N, 19.1134 E)
- Area around ca. 160 m Sudół stream stretch near Kraków city (50.0999 N, 19.9027 E)

The water temperature of the rivers was measured directly along their course in each case. The results were constant for the entire section and did not depend on chainage. Table 1 provides details of the locations, dates, conditions of the surveys, and measured temperatures.

Table 1. Details of the surveys carried out.

Alias	Location	Time	Conditions	Water temperature
20211215_kocinka_rybna	Kocinka, Rybna	15.12.2021 12:20	Fog, snow cover	4.6 °C
20220118_kocinka_rybna	Kocinka, Rybna	18.01.2022 14:55	Snow cover, total cloud cover	2.6 °C
20220325_kocinka_rybna	Kocinka, Rybna	25.03.2022 07:30	No cloud cover	5.6 °C
20221220_sudol_krakow	Sudół, Kraków	20.12.2022 11:20	Moderate cloud cover	2.0 °C
20230111_kocinka_grodzisko	Kocinka, Grodzisko	11.01.2023	No cloud cover	4.2 °C

2.2. Algorithm

2.2.1. Vignette correction

The correction of the vignette effect was conducted using the “single image” method [12]. ‘Single image’ means that the algorithm tries to model the correction of the vignette effect only on the basis of the currently processed image and does not have any auxiliary data available (e.g., an image of a homogeneous target with a clearly visible vignette effect). It was implemented by translating the MATLAB code available at https://github.com/GUOYI1/Vignetting_corrector into Python. Several changes have been made to adapt the algorithm to work with thermal images. Standard images are represented with floating point values from 0 to 1, or integers from 0 to 255. The original

implementation of the algorithm assumed this data format, so it had to be modified to work on unconstrained float values. Thermal image values can also be negative. The vignette correction algorithm works on logarithms of pixel values, so temperatures are converted to Kelvins in order to avoid the logarithm of 0. The vignette correction algorithm tends to increase the brightness of the photo. For standard photos this is not a problem, but for thermal images, where increasing the brightness will result in a bias in the temperature reading, this cannot be accepted. We assume that images with vignette effect occurring contain correct temperature in their central part. Therefore, we can use the central part of the photo before correction as a reference level for the photo obtained after applying the vignette correction algorithm. The bias is compensated for by subtracting the average difference between the central areas of the images before and after vignette correction. While the vignette effect mainly affects the edge part of the image, the central area size is defined as a rectangle with sides twice smaller than the whole image.

2.2.2. Georeferencing

The georeferencing algorithm is largely based on the aerial photos stitching solution proposed by Luna Yue Huang [13]. Modifications in our method consists of direct georeferencing to Universal Transverse Mercator (UTM) coordinate system.

The first part of the georeferencing procedure is initial georeferencing of images based on EXIF metadata. These contain information about the geographic coordinates, yaw, and altitude of the drone camera at the time the image was taken. This information, along with the angle of field of view taken from the camera specifications, allows estimation of an affine transformation parameters of translation (v_x, v_y) , scale (s_x, s_y) , and counter clockwise rotation (θ) that allows to embed the image in an UTM coordinate system. This is not an accurate estimation, due to the low accuracy of the UAV's GPS geolocation and high sensitivity to the shift of the camera viewing area under the influence of wind blows. In this estimation the scale s_x and s_y parameters are equal, and the shear (c_x, c_y) coefficients are 0. The transformation parameters allow us to obtain the transformation matrix A based on Equation 1.

$$A = \begin{pmatrix} \cos \theta & -\sin \theta & v_x \\ \sin \theta & \cos \theta & v_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & c_x & 0 \\ c_y & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} s_x \cos \theta - c_y \sin \theta & c_x \cos \theta - s_y \sin \theta & v_x \\ c_y \cos \theta + s_x \sin \theta & s_y \cos \theta - c_x \sin \theta & v_y \\ 0 & 0 & 1 \end{pmatrix}$$

Based on the initial georeferencing, pairs of overlapping images are found. To ensure that all possible pairs are found even when errors in the initial georeferencing may indicate a lack of overlapping, footprints are expanded (buffered) by a dilation operation by the given padding value. Each pair of images is aligned and a relative transformation matrix A_R is found by a well-established in the field of image vision stitching method: i) finding corresponding points (keypoints) in both images using scale-invariant feature transform (SIFT) [14], ii) matching keypoints using fast approximate nearest neighbour search (FLANN) [15], and iii) estimation of best transformation matrix using random sample consensus (RANSAC) [16]. Relative transformations are verified in two stages: i) since we assume that the images are taken from the same altitude, the scaling factor in the relative transformation of two overlapping images must be approximately equal to 1. If the value of the scaling factor for relative transformations is outside the range 0.9 – 1.1, the relative transformation is considered incorrect. ii) as all images are taken in nadir orientation, the relative transformations should not perform a shearing operation. When there is no shear, the absolute values of the relative transformation matrix A_R satisfy the following condition: $|A_{R11}| = |A_{R22}|$ and $|A_{R12}| = |A_{R21}|$. If deviation between these values is greater than 0.1, the relative transformation is considered to be incorrect. If the relative transformation is found to be incorrect, according to Luna Yue Huang solution, an attempt is made to establish a relative transformation using the same pair of images, but resized to twice lower resolution. Such a procedure allows you to obtain a different keypoints using

for scale determination. If the relative transformation obtained in this way again fails to pass verification, the pair of photos is discarded.

Based on the obtained pairs, an undirected graph was constructed [17], where each node is an aerial photo, and each edge is a valid relative transformation between a pair of photos. The connected components are then extracted from the graph. A connected component in graph theory is a group of vertices in a graph that are all directly or indirectly connected to each other. Only images (nodes) from a connected component containing the largest number of nodes are used for further processing.

Iterating over each pair of images, the coordinates of the corners of the second image are calculated in the pixel coordinate system of the first image using equation 2. This idea is also visualised in Figure 1.

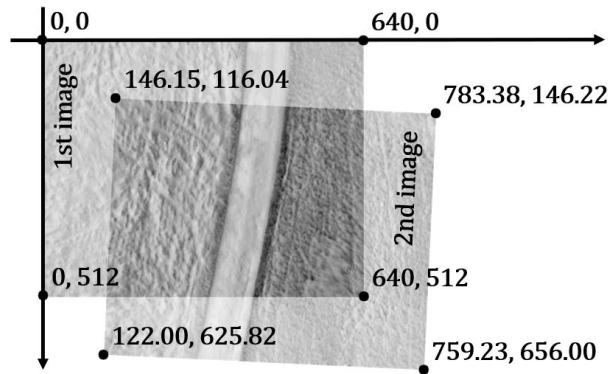


Figure 1. Alignment of an example pair of images with dimensions of 640×512 pixels. The coordinates of the marked corners are expressed in the pixels coordinate system of the first image.

$$p_{1st} = A_R \cdot p_{2nd}, \quad (2)$$

where:

A_R – relative transformation matrix between 2nd image and 1st image pixel coordinate system,

p_{2nd} – coordinates of corner point expressed in 2nd image pixel coordinate system,

p_{1st} – coordinates of the same point expressed in 1st image pixel coordinate system.

Optimisation of georeferencing of all images involves tuning the absolute geographic transformation parameters (v_x , v_y , s_x , s_y , θ) to make the relative transformations recovered from them as close as possible to the relative transformations obtained earlier using alignment of each pair separately. Recovery of the relative transformation matrix \widehat{A}_R from the absolute geographic transformation matrices \widehat{A}_{1st} and \widehat{A}_{2nd} of two georeferenced images is obtained according to Equation 3.

$$\widehat{A}_R = \widehat{A}_{1st}^{-1} \cdot \widehat{A}_{2nd}, \quad (3)$$

c_x and c_y shear parameters are not tuned during the optimization. Although tuning of these parameters further improves the matching between pairs of photos, it also introduces an unacceptable shearing of the entire mosaic of photos.

Similarly to equation 2, the coordinates of points expressed in the pixel reference system of the 2nd photo can be converted to coordinates expressed in the reference system of the 1st image using equation 4.

$$\widehat{p}_{1st} = \widehat{A}_R \cdot p_{2nd}, \quad (4)$$

where:

\widehat{A}_R – relative transformation matrix between 2nd image and 1st image pixel coordinate system,

p_{2nd} – coordinates of point expressed in 2nd image pixel coordinate system,

\widehat{p}_{1st} – coordinates of the same point expressed in 1st image pixel coordinate system.

Optimisation is performed using the gradient descent method with the loss function \mathcal{L} that consist of two components \mathcal{L}_R and \mathcal{L}_A . Component \mathcal{L}_R is the mean Euclidean distance between the

points at the corners of the images obtained from the relative transformations recovered from the tuned absolute geographic transformations and the points from the corners of the images from the relative transformations obtained during the pair alignment.

$$\mathcal{L}_R = \frac{1}{N_P} \sum_{i=1}^{N_P} \frac{1}{4} \sum_{j=1}^4 \|p_{1st,i,j} - \widehat{p_{1st,i,j}}\|, \quad (5)$$

where:

N_P – number of pairs,

$p_{1st,i,j}$ – point of j-th corner of the i-th pair 2nd image expressed in the 1st image pixel coordinate system estimated from pairs alignment,

$\widehat{p_{1st,i,j}}$ – point of j-th corner of the i-th pair 2nd image expressed in the 1st image pixel coordinate system estimated using relative transformation recovered from absolute geographic transformations tuned during optimization.

Component \mathcal{L}_A is a mean Euclidean distance between geographic centroids of tuned image footprints and geographic centroids of image footprints obtained during initial georeferencing using EXIF data.

$$\mathcal{L}_A = \frac{1}{N_I} \sum_{i=1}^{N_I} \|p_i - \widehat{p_i}\|, \quad (6)$$

where:

N_I – number of images,

p_i – point of centroid of i-th image obtained from EXIF data expressed in geographic coordinate system,

$\widehat{p_i}$ – point of centroid of i-th image obtained from absolute geographic transformation tuned during optimization.

By minimizing the \mathcal{L}_R factor, the optimized absolute geographic transformations are adjusted so that the retrieved from them relative transformations between pairs of images are as close as possible to the relative transformations previously obtained separately for each pair in the alignment process. Minimizing the \mathcal{L}_a factor ensures that the entire mosaic will not move in any direction during the optimization. Equation 7 is the final formula for the \mathcal{L} loss function. To minimise the impact on the relative matching of images using \mathcal{L}_R component, the \mathcal{L}_A component is multiplied by a factor of 10^{-6} . The factor value was selected experimentally by qualitative assessment of results tested for different values.

$$\mathcal{L} = \mathcal{L}_R + 10^{-6} \mathcal{L}_A, \quad (7)$$

2.2.3. Thermal calibration

In order to minimise the temperature bias between overlapping images using Gradient Descent, a consistent dataset has to be prepared. If the reduction of the vignette effect has not yielded sufficient results and the photos overlap is large enough, the areas of the photos near the edges where the vignette effect affects the temperature images the most can be truncated (Figure 2).

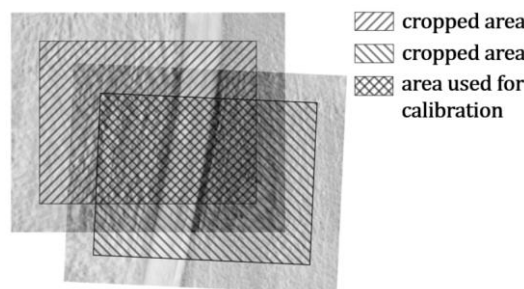


Figure 2. Example of cropped areas and the common area of a pair of images used for calibration.

For the two images from each pair, only the common part is retained, as well as a mask that allows us to reproduce the irregular shape of the cutout from the rectangular array. Temperature cutouts of both images and mask are resized to array of 32×32 pixels. During resizing, the temperatures are interpolated using the bilinear method, and the mask is interpolated using the nearest neighbours method. The example result of the common clip is shown at Figure 3. The example corresponds to the images shown in Figures 1 and 2. The rotation is due to the transformation to an absolute geographic datum. The lack of the preservation of aspect ratio is due to the need for scaling to fill the whole area of square array of 32×32 .

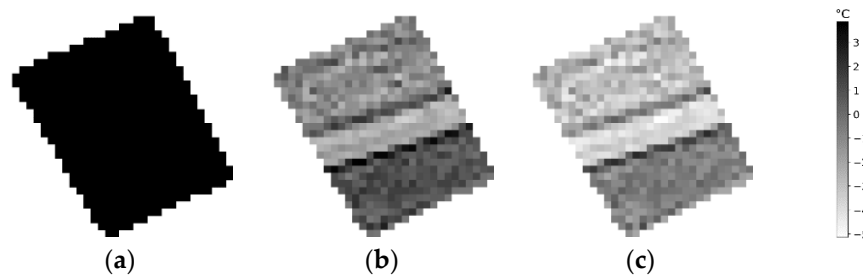


Figure 3. Example components of the dataset sample: a) binary mask, b) first temperature image, c) second temperature image.

By resizing the slices to 32×32 pixels, it was possible to create a coherent dataset of three arrays (masks and pair of images common part) of size $N_p \times 32 \times 32$ (N_p is number of pairs), which could then be used in optimization using the gradient method.

Optimisation of the temperature values involves tuning the b parameter that is used for the offset of each image according to equation 8.

$$\hat{v} = v + b, \quad (8)$$

where:

\hat{v} – temperature value with applied calibration,

v – temperature value before calibration,

Optimisation uses a loss function \mathcal{L} that consists of two components: \mathcal{L}_B and \mathcal{L}_M . Component \mathcal{L}_B is pixelwise mean squared error between calibrated temperature values of first and second cutout images from given pair. It is calculated with Equation 8.

$$\mathcal{L}_B = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{1}{32} \sum_{j=1}^{32} \frac{1}{32} \sum_{k=1}^{32} (\widehat{v_{1st,t,j,k}} - \widehat{v_{2nd,t,j,k}})^2, \quad (9)$$

where:

N_p – number of pairs,

$\widehat{v_{1st,t,j,k}}$ – value of calibrated first image in i -th pair at pixel coordinate (j, k) ,

$\widehat{v_{2nd,t,j,k}}$ – value of calibrated first image in i -th pair at pixel coordinate (j, k) ,

\mathcal{L}_M is absolute difference between mean of all uncalibrated images and mean of calibrated images. It is calculated with Equation 9.

$$\mathcal{L}_M = |\mu - \hat{\mu}|, \quad (10)$$

where:

μ – mean value of uncalibrated images,

$\hat{\mu}$ – mean value of calibrated images.

By minimizing the \mathcal{L}_B component, the bias between the temperature values in the overlapping images is minimized. The \mathcal{L}_M component ensure that during optimization the whole mosaic will not drift from initial mean value. The total loss is calculated using equation 11. To minimise the impact on bias correction using the \mathcal{L}_B component, the \mathcal{L}_M component was multiplied by factor of 10^{-2} . The factor value was selected experimentally by qualitative assessment of results tested for different values.

$$\mathcal{L} = \mathcal{L}_B + 10^{-2} \mathcal{L}_M, \quad (11)$$

2.3. Final processing and result analysis

The rivers in the measured areas have homogenous temperatures throughout the surveyed sections and a well-defined emissivity value of 0.95. Therefore, they can be used to verify the proposed method. The sets of both the original and bias-corrected images were offset (the same offset value for all images from the given set) so that the average water body temperature read from the thermal mosaic was equal to the measured reference water temperature. We decided on this because the temperatures returned by the DJI Thermal SDK may differ significantly from the actual temperatures, it is straightforward correction and makes it easier to compare measurements from before and after calibration.

3. Results

3.1. Vignette correction

A 'single image' algorithm was used to correct the vignette effect. For this reason, it is prone to misinterpretations of the image and the resulting erroneous correction of the vignette effect. Figure 4 shows an example of correct vignette effect correction. In this case, the standard deviation of the image as a result of the vignette effect correction has decreased, which manifests itself as a narrowing of the histogram of pixel values. Figure 5 shows an incorrect correction of the vignette effect, where it is likely that the river near the left edge of the image caused the misinterpretation. In this case, the correction resulted in an increase in the standard deviation of the photo, which manifests itself as a widening of the histogram of pixel values.

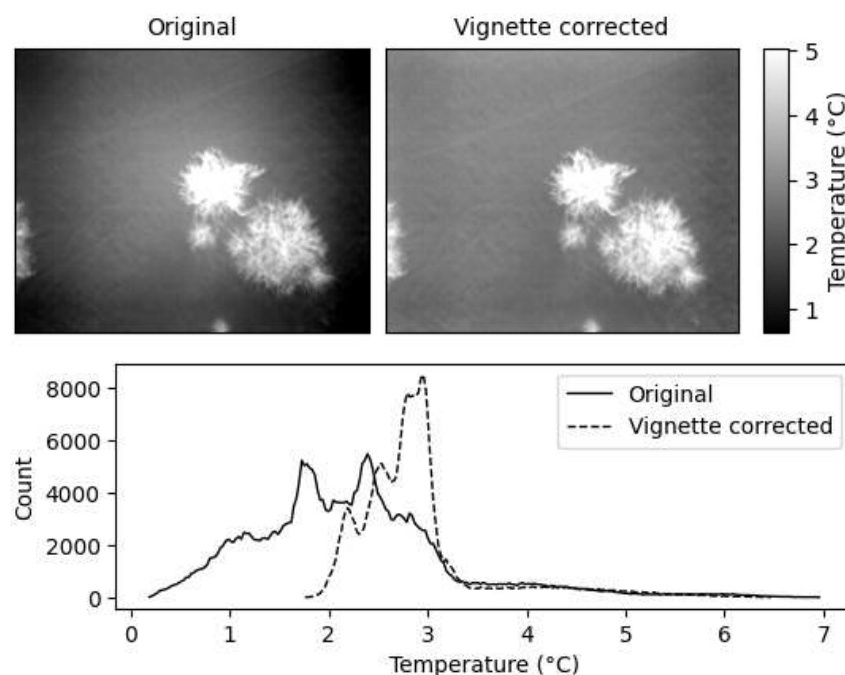


Figure 4. Example of successful vignette correction.

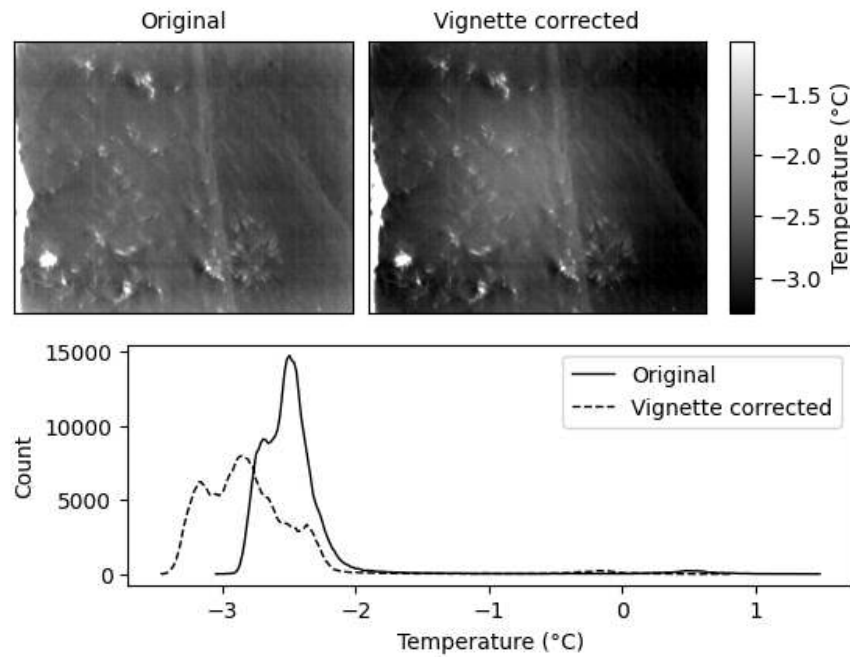


Figure 5. Example of unsuccessful vignette correction.

To quantitatively investigate the effectiveness of the devignetting algorithm, it was checked for how many images a reduction in standard deviation was obtained as a result of devignetting. For 74.4% of the images (2261 out of 3037 total images) there was a reduction in the standard deviation. For 25.6% of the images (776 out of 3037 total images) there was an increase in the standard deviation. The average reduction in standard deviation was -0.07°C while the average increase in standard deviation was 0.01°C . The detailed distribution of the changes in standard deviation as a result of the vignette effect correction is shown in the histogram in Figure 6.

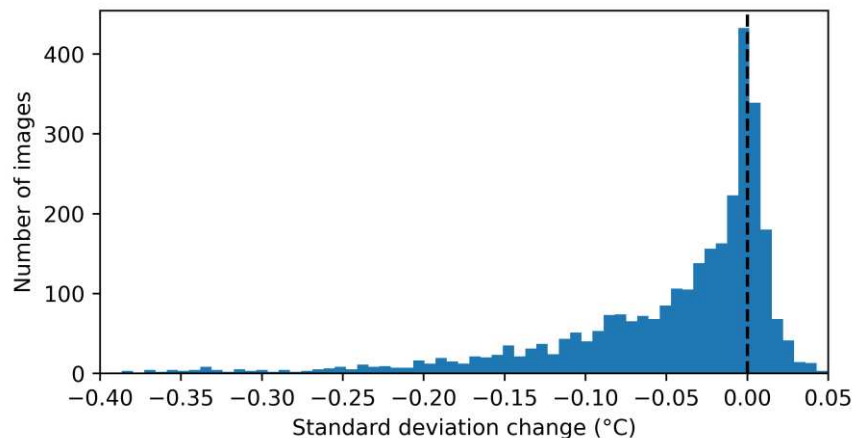


Figure 6. Change in the standard deviation of the images as a result of the devignetting algorithm. The dashed line is drawn at zero.

3.2. Visual assessment

Figures 7–11 show parts of thermal image mosaics before and after calibration for each case study. These are not orthomosaics - they were created straightforwardly by displaying all georeferenced images. Where the images overlap, a later image from the collection is displayed.

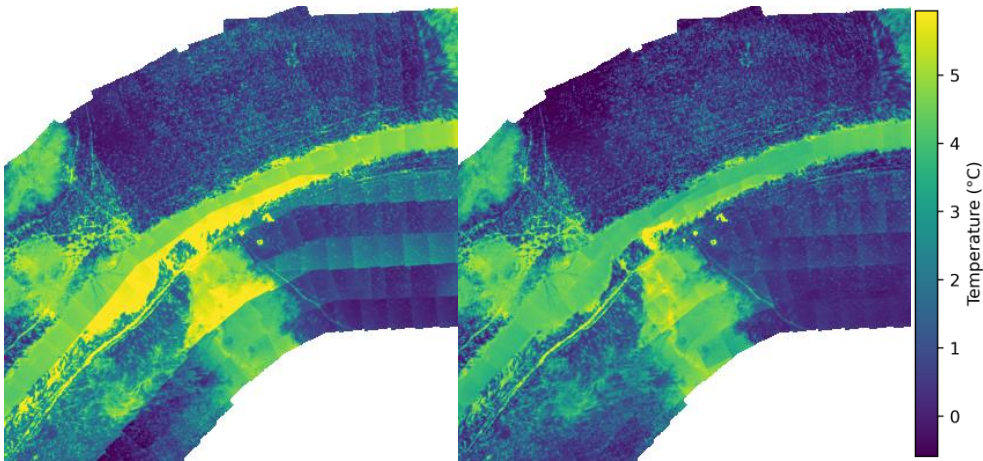


Figure 7. 20211215_kocinka_rybna – fragment of mosaic before (left) and after (right) the calibration.

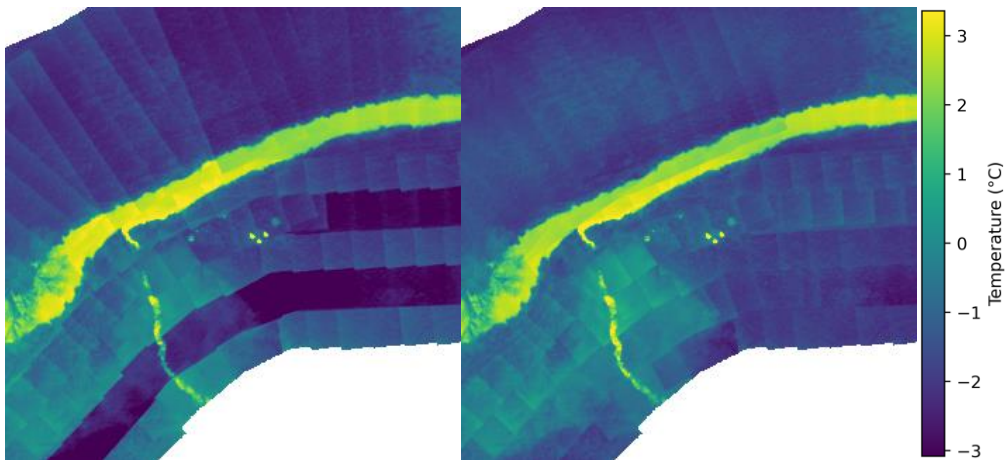


Figure 8. 20220118_kocinka_rybna – fragment of mosaic before (left) and after (right) the calibration.

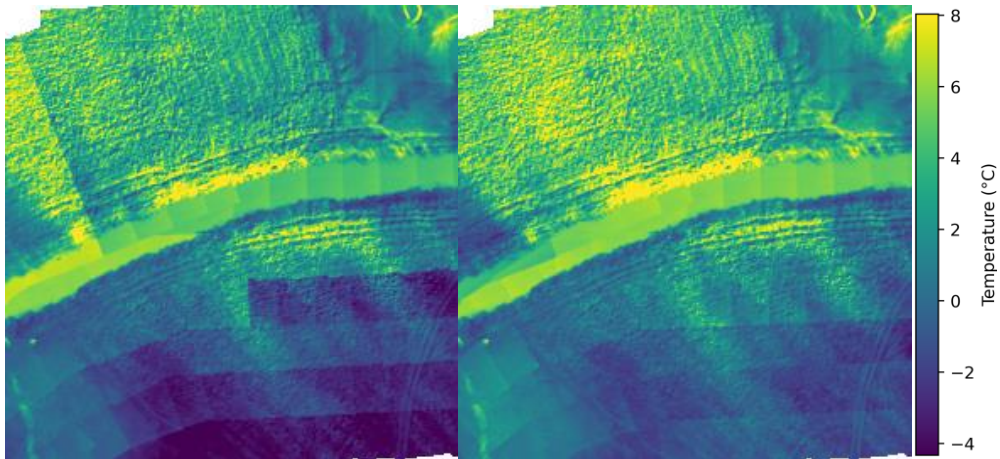


Figure 9. 20220325_kocinka_rybna – fragment of mosaic before (left) and after (right) the calibration.

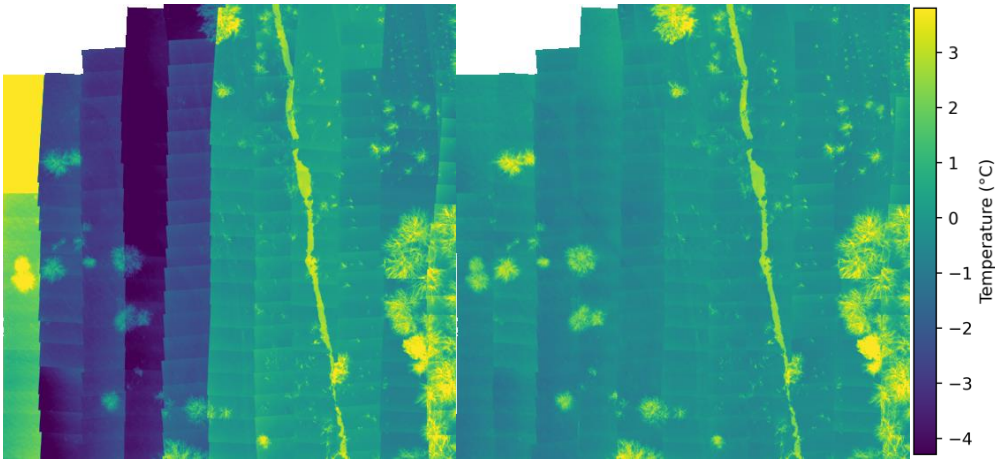


Figure 10. 20221220_sudol_krakow – fragment of mosaic before (left) and after (right) the calibration.

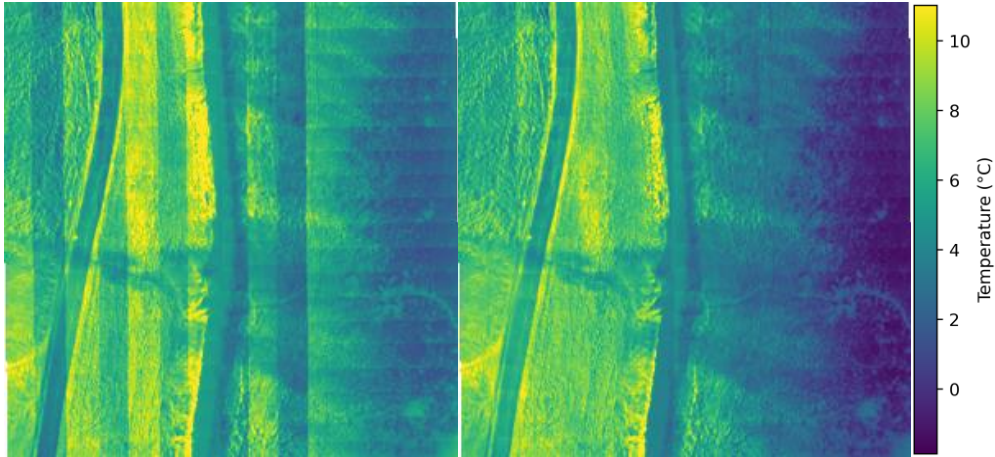


Figure 11. 20230111_kocinka_grodzisko – fragment of mosaic before (left) and after (right) the calibration.

3.3. Waterbody temperature

Table 2 summarizes the obtained RMSE and MAE of stream temperature measurements using uncalibrated and calibrated thermal images. Both calibrated and uncalibrated images were subjected to the offset described in section 2.3.

Table 2. Comparison of root mean squared error and mean absolute error of stream temperature measurement using uncalibrated and calibrated thermal images.

Case study	RMSE	RMSE	MAE	MAE
	Uncalibrated	Calibrated	Uncalibrated	Calibrated
20211215_kocinka_rybna	0.679375	0.310173	0.539997	0.249083
20220118_kocinka_rybna	0.958521	0.528135	0.579392	0.337545
20220325_kocinka_rybna	1.229638	0.93813	0.894125	0.664885
20221220_sudol_krakow	1.099734	0.703925	0.831873	0.519501
20230111_kocinka_grodzisko	0.96338	0.61609	0.815541	0.458868

Figure 12 show the temperatures sampled along the river centreline from the uncalibrated and the calibrated images. There are multiple measurements for a single chainage value, as a given point may be present in multiple overlapping images. Peaks in the temperature readings that have not been compensated by calibration are sampled on vegetation overhanging the water surface.

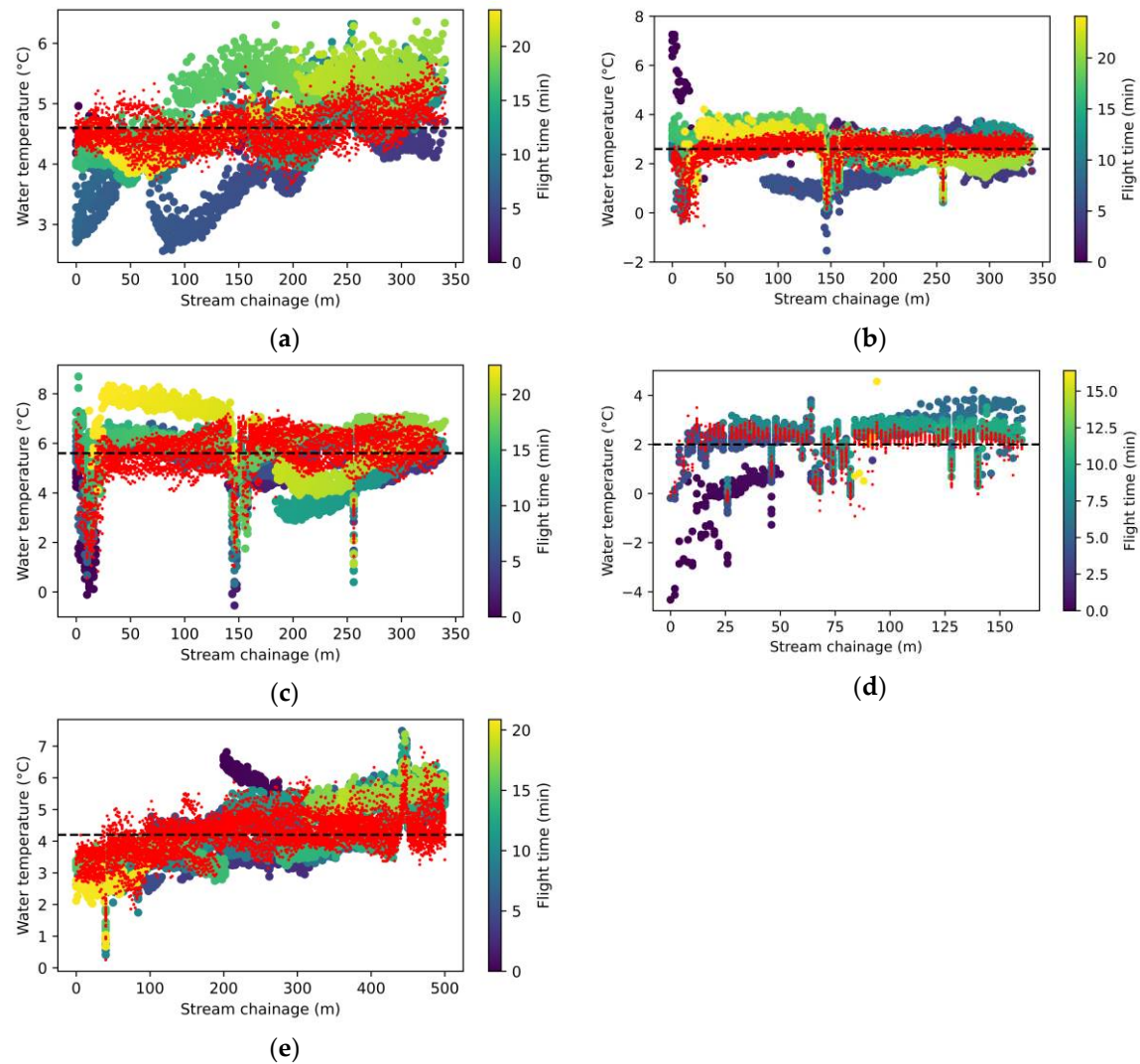


Figure 12. Temperatures sampled along the river centreline from the uncalibrated (flight time colour scale large points) and the calibrated images (red small points) for each case study. The dashed line is the actual water temperature. (a) 20211215_kocinka_rybna, (b) 20220118_kocinka_rybna, (c) 20220325_kocinka_rybna, (d) 20221220_sudol_krakow, (e) 20230111_kocinka_grodzisko.

4. Discussion

4.1. Devignetting

The single-image vignette correction method has proven successful when applied to thermal images. Although there was an increase in the standard deviation for about a quarter of the images. This was a minor increase, as on average a seven times greater reduction in standard deviation was recorded for the rest of the three-quarters of the images.

4.2. Visual assessment of mosaics

A visual comparison of the mosaics before and after calibration shows an improvement in temperature fluctuations, particularly between successive flight passages. The algorithm also handles the correction of undervalued images taken at the beginning of the flight, when the camera is warming up. Correction of the images from the case study 20211215_kocinka_rybna and 20220118_kocinka_rybna has allowed us to see the warmer tributary of the stream, which is fed by warmer groundwater (Figures 7 and 8 after calibration).

4.3. Measurement precision and accuracy

The proposed solution based on minimising the bias between overlapping images significantly increased the precision of the water temperature measurements taken in all case studies. Despite the improvement in calibration precision, the accuracy can still be poor due to imperfections of the Thermal SDK library provided by the manufacturer, which was used for conversion of raw data to temperature values. However, an increase in accuracy can be easily achieved by an additional offset applied to all images (see Section 2.3) if any reference points are available for the study area. After applying this offset, the average root mean squared error was reduced by 39.0% and the mean absolute error by 40.5%.

4.4. Applications

Previous methods of reducing adverse phenomena occurring in uncooled thermal cameras have relied on calibration under controlled laboratory conditions and required non-standard instrumentation such as reference black bodies or custom UAV attachments. For this reason, they require an additional effort of specialised operator and cannot be automated. On the other hand, our proposed solution requires minimal effort as it only requires data collected during a standard UAV flight and is highly automated making it possible to use by a non-specialised operator. This also makes the solution ready to be implemented as part of a photogrammetric software workflow.

Author Contributions: “Conceptualization, R.S., M.Z., P.W.; methodology, R.S.; field work, R.S., .Z., P.W.; software, R.S.; validation, R.S.; formal analysis, R.S.; investigation, R.S.; resources, M.Z. P.W., R.S.; data curation, R.S., M.Z.; writing—original draft preparation, R.S.; writing—review and editing, P.W., M.Z.; visualization, R.S.; supervision, P.W., M.Z.; project administration, P.W., M.Z.; funding acquisition, P.W., M.Z. All authors have read and agreed to the published version of the manuscript.”.

Funding: Research was partially funded by the National Science Centre, Poland, project WATERLINE (2020/02/Y/ST10/00065), under the CHISTERA IV programme of the EU Horizon 2020 (Grant no 857925) and the "Excellence Initiative - Research University" program at the AGH University of Science and Technology.

Data Availability: The data presented in this study are openly available in Zenodo repository at <https://doi.org/10.5281/zenodo.8359964>.

Acknowledgments: This research was supported in part by PL-Grid Infrastructure.

Conflicts of Interest: The authors declare no conflict of interests.

Source Code: The Python implementation source code is available in form of Jupyter notebooks at the GitHub repository <https://github.com/radekszostak/aerial-thermal-tuner>. Notes on implementation are included in the appendix A.

Appendix A

The algorithm was implemented in Jupyter Notebooks using the Python programming language. The processing solution includes 5 notebooks responsible for preliminary images conversion, vignette correction, georeferencing of images, calibration of temperature values and offsetting of all images (*1_conversion.ipynb*, *2_devignetting.ipynb*, *3_georeferencing.ipynb*, *4_calibration.ipynb*, respectively). They can be run on any operating system thanks to the Docker containerisation.

Conversion notebook

First, the images must be converted from the camera manufacturer's specific format (usually a JPG image with RGB channels providing temperature visualisation) to a single-channel TIFF file containing temperature values. For this study, the DJI Thermal SDK library provided by the thermal camera manufacturer was used for extraction of temperature arrays from manufacturer-specific JPG file format. This library requires values of flight altitude, air humidity, surface emissivity,

and surface reflection temperature. The extracted temperatures are saved to a TIF file. The EXIF data is copied unchanged from the input JPG file to the resulting TIF file.

Georeferencing notebook

For initial georeferencing, longitude (λ), latitude (φ), altitude (z) and rotation (θ) data were extracted from the images using the *exiftool* software. The camera's diagonal angle of view (α) was taken from the camera documentation. The length of the image footprint diagonal (l_d) expressed in meters was calculated with $l_d = 2z \tan(0.5\alpha)$ formula. The horizontal and vertical footprint length was calculated using $l_h = l_d(p_h/\sqrt{p_h^2 + p_v^2})$ and $l_v = l_d(p_v/\sqrt{p_h^2 + p_v^2})$ formulas respectively, where p_h and p_v are width and height of image expressed in number of pixels (in this case study $p_h = 640$ and $p_v = 512$). The longitude and altitude were converted into x , y coordinates of the UTM system using the *pyproj* library. These are the coordinates of the centroid of the footprint of the photo. Footprint was initialized using *Polygon* object of the *shapely* library with four corner points: $(x - 0.5l_h, y - 0.5l_v)$, $(x + 0.5l_h, y - 0.5l_v)$, $(x + 0.5l_h, y + 0.5l_v)$, $(x - 0.5l_h, y + 0.5l_v)$. It was then rotated by θ around the centroid using *affinity.rotate* function from *shapely* library to obtain final footprint.

A modified variant of the transformation matrix was used in the implementation to take into account that images expressed as arrays have the y-axis pointing downward. The modified transformation matrix is similar to standard transformation matrix with clockwise rotation; however, it uses clockwise rotation and bottom-right pixel coordinates are used as v_x and v_y parameters. The modified version of the transformation matrix used in the implementation of the algorithm is presented in equation A1.

$$A = \begin{pmatrix} \cos \theta & \sin \theta & v_x \\ \sin \theta & -\cos \theta & v_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & c_x & 0 \\ c_y & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (A1)$$

$$= \begin{pmatrix} s_x \cos \theta + c_y \sin \theta & c_x \cos \theta + s_y \sin \theta & v_x \\ s_x \sin \theta - c_y \cos \theta & c_x \sin \theta - s_y \cos \theta & v_y \\ 0 & 0 & 1 \end{pmatrix},$$

The modified transformation matrix was composed with v_x and v_y parameters being coordinates of the bottom-right footprints corner, s_x and s_y parameters being pixel size calculated with $s_x = s_y = l_d/\sqrt{p_h^2 + p_v^2}$, θ parameter equal to negative yaw value extracted from EXIF data (as counterclockwise rotation is needed), and shear c_x and c_y parameters equal to 0.

In order to optimize the process of finding matches of overlapping pairs of photos, candidates are found that are potentially overlapping. The process of finding candidates consists of dilating all footprints by 5m and then finding pairs of overlapping dilated footprints. Dilation is necessary, because due to inaccurate initial georeferencing some of footprints may indicate no overlap even if corresponding photos contain common area. Then, with the help of the CV2 library, an attempt is made to stitch a given image with each of the potential neighbours. Stitching procedure consist of several steps using functions from *cv2* library: i) detecting features and computing descriptors with SIFT algorithm using *detectAndCompute* function, ii) descriptor matching using *knnMatch* function from *FlannBasedMatcher* class object, iii) filtering out incorrect matches using Lowe's ratio test with ratio value equal 0.7, iv) using correct matches with *estimateAffinePartial2D* function to estimate relative transformation matrix between given image and potential overlapping neighbor. If the number of matching points used for calculation of the transformation matrix is greater than or equal to 8, the pair of images and their relative transformation matrix are qualified for further processing. At this stage, using a relative transformation matrix, the coordinates of the corners of the stitched image in the relative reference system of the other image of the pair are also calculated.

To further ensure that the stitched pairs are correct, we filter out pairs where the scaling factor in the relative transformation matrix is outside the range of 0.8 to 1.2, as we assume that the images were taken from the same altitude, so the relative scaling factor must be equal to 1.

All stitched pairs of images that remain up to this point are interpreted as undirected graphs using the *networkx* library. Using the *connected_components* function, the connected component that contains the largest number of nodes (images) is extracted. It is also possible to visualise the stitched pairs and connected components as a graph with the geographical position of the nodes preserved. An example graph visualisation is shown in Figure A1.

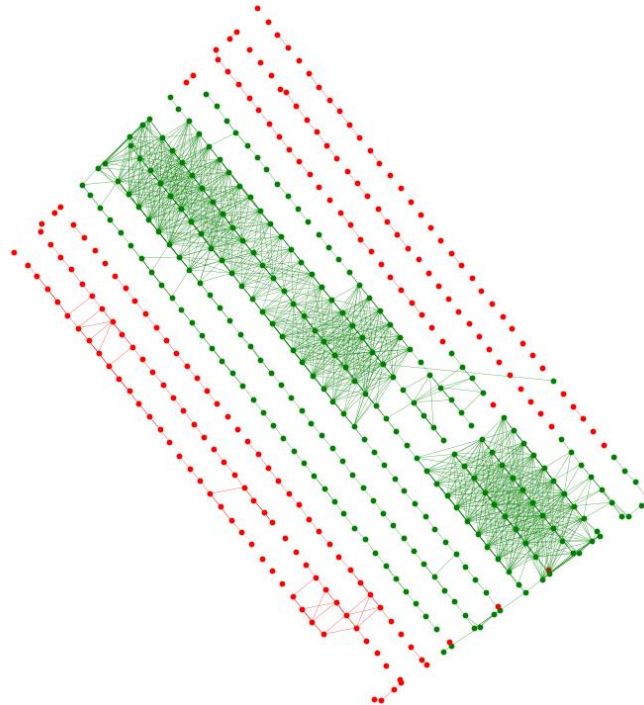


Figure A1. Graph visualization with preserved relative geographical positions of nodes and largest connected component marked in green. Green nodes are used for georeferencing optimization, red nodes are discarded. Example for Sudól case study.

The data prepared up to this point is loaded into the tensor objects of the *pytorch* library. Values of these tensors relate either to images or image pairs. Transformation parameters v_x , v_y , θ , s_x , s_y , c_x , c_y are stored in separate tensors of length equal number of images. These tensors will be tuned during the gradient descent optimization, so they are initialized with parameter *requires_grad=True*. A short optimisation time is ensured by using the ReduceLROnPlateau learning rate scheduler.

Georeferencing notebook

Footprints are read from previously generated GeoTIFF files. Then, based on them, pairs are found whose common part area is not less than the value defined as *MIN_INTERCESTION_AREA*. Samples that are the common part in pairs of images are scaled to a low resolution, defined as *CLIP_SIZE*. The data set is loaded into the tensors and gradient descent optimisation is performed.

References

1. Berni, J.A.J.; Zarco-Tejada, P.J.; Suarez, L.; Fereres, E. Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring From an Unmanned Aerial Vehicle. *IEEE Transactions on Geoscience and Remote Sensing* **2009**, doi:10.1109/tgrs.2008.2010457.
2. Yao, H.; Qin, R.; Chen, X. Unmanned Aerial Vehicle for Remote Sensing Applications—A Review. *Remote Sensing* **2019**, *11*, 1443, doi:10.3390/rs11121443.
3. Lagüela, S.; Díaz-Vilariño, L.; Roca, D.; Lorenzo, H. Aerial Thermography from Low-Cost UAV for the Generation of Thermographic Digital Terrain Models. *Opto-Electronics Review* **2015**, *23*, doi:10.1515/oere-2015-0006.

4. Messina, G.; Peña, J.M.; Vizzari, M.; Modica, G. A Comparison of UAV and Satellites Multispectral Imagery in Monitoring Onion Crop. An Application in the 'Cipolla Rossa Di Tropea' (Italy). *Remote Sensing* **2020**, *12*, 3424, doi:10.3390/rs12203424.
5. Aragon, B.; Phinn, S.R.; Johansen, K.; Parkes, S.; Malbeteau, Y.; Al-Mashharawi, S.; Alamoudi, T.S.; Andrade, C.F.; Turner, D.; Lucieer, A.; et al. A Calibration Procedure for Field and UAV-Based Uncooled Thermal Infrared Instruments. *Sensors* **2020**, doi:10.3390/s20113316.
6. Kelly, J.; Kljun, N.; Olsson, P.-O.; Mihai, L.; Liljeblad, B.; Weslien, P.; Klemedtsson, L.; Eklundh, L. Challenges and Best Practices for Deriving Temperature Data from an Uncalibrated UAV Thermal Infrared Camera. *Remote Sensing* **2019**, *11*, 567, doi:10.3390/rs11050567.
7. Kusnierek, K.; Korsæth, A. Challenges in Using an Analog Uncooled Microbolometer Thermal Camera to Measure Crop Temperature. *International Journal of Agricultural and Biological Engineering* **2014**.
8. Yuan, W.; Hua, W. A Case Study of Vignetting Nonuniformity in UAV-Based Uncooled Thermal Cameras. *Drones* **2022**, doi:10.3390/drones6120394.
9. Ribeiro-Gomes, K.; Hernández-López, D.; Ortega, J.; Ballesteros, R.; Poblete, T.; Moreno, M. Uncooled Thermal Camera Calibration and Optimization of the Photogrammetry Process for UAV Applications in Agriculture. *Sensors* **2017**, *17*, 2173, doi:10.3390/s17102173.
10. Lin, D.; Maas, H.-G.; Westfeld, P.; Budzier, H.; Gerlach, G. An Advanced Radiometric Calibration Approach for Uncooled Thermal Cameras. *Photogrammetric Record* **2018**, doi:10.1111/phor.12216.
11. Mesas-Carrascosa, F.J.; Pérez-Porras, F.J.; Larriva, J.E.M. de; Frau, C.M.; Agüera-Vega, F.; Carvajal-Ramírez, F.; Martínez-Carricondo, P.; García-Ferrer, A. Drift Correction of Lightweight Microbolometer Thermal Sensors On-Board Unmanned Aerial Vehicles. *Remote Sensing* **2018**, doi:10.3390/rs10040615.
12. Yuanjie Zheng; Lin, S.; Kambhamettu, C.; Jingyi Yu; Sing Bing Kang Single-Image Vignetting Correction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 2243–2256, doi:10.1109/TPAMI.2008.263.
13. Huang, L.Y. Luna983/Stitch-Aerial-Photos: Stable v1.1 2020.
14. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* **2004**, *60*, 91–110, doi:10.1023/B:VISI.0000029664.99615.94.
15. Muja, M.; Lowe, D.G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In Proceedings of the Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (VISIGRAPP 2009) - Volume 1: VISAPP; SciTePress, 2009; pp. 331–340.
16. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395, doi:10.1145/358669.358692.
17. John, C.; Allan, H.D. *A First Look at Graph Theory*; Allied: New Delhi, India, 1995;

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.