Ref. Ares(2019)1382596 - 28/02/2019

Project Number: 774571
Start Date of Project: 2017/11/01
Duration: 48 months

Type of document D.4.3 – V1.0

Aerial Image Processing Pipeline

Dissemination level	PU
Submission Date	2019-02-28
Work Package	WP4
Task	T4:5
Type	Report
Version	1.0
Author	Chizhang Gong, Sebastian Lemprecht
Approved by	Emanuele Garone + PMC

Executive Summary

This document describes the details of the UAV image processing pipeline to generate multiband aerial orthophoto mosaic raster images. These data sets are required for *Task 4.6 Water Stress Measurement*, *Task 4.7 Pest and Disease Detection*. The following aspects are presented:

1. Selection of the sensors
2. General introduction of the pipeline
   2.1 The methods and implementation of image pre-procesing
   2.2 The methods and implementation of orthophoto mosaic computation
3. Test results of the pipeline

Table of Content

Abbreviations and Acronyms

API	Application programming interface
CWSI	Crop water stress index
DEM	Digital elevation model
DN	Digital Number
DSM	Digital surface model
FOV	Field of view
NDVI	Normalized differential vegetation index
NIR	Near-infrared
PRI	Photochemical reflectance index
RTK	Real time kinematic
SCADA	Supervisory control and data acquisition
SfM	Structure from motion
UAV	Unmanned aerial vehicle
WP	Work package

1 Introduction

Task 4.5 focuses on developing an automatic pipeline to process the raw aerial images into orthophoto mosaic maps. The aim of this task is to provide geo accurate and band-wise aligned orthophoto mosaics that *Task 4.6 Water Stress Measurement* and *Task 4.7 Pest and Disease Detection* can work on.

The pipeline consists of two main parts:

- Image pre-processing
- Orthophoto mosaic generation via bundle adjustment

Optical digital images directly collected from fields are subject to a variety of unwanted errors introduced by camera optics, environmental impacts and digital conversion. To receive meaningful comparable measurements, a pre-processing phase of field collected images before the orthophoto mosaic is by all means necessary. This requires a series of laboratory and field calibration to take place, whose methods are largely identical to those in *Deliverable 4.1 (D.4.1)*.

To compute orthogonally projected raster image from processed images, a geometric relation between 2D images and 3D real world is calculated. The so called bundle adjustment process is individually well understood [1] to tackle the problems of large systems of normal equations solution or position error detection and elimination. In this pipeline, these solutions are constructed from *Agisoft© Metashape*, which provides a Python application programming interface (API) for *Linux* system, allowing the smooth software integration in the SCADA architecture.

2 Selection of the sensors

A range of impacts on the vegetation can be detected by remote sensing techniques in specific spectral ranges due to their corresponding biological responses. For these spectral analyses, the vegetation index is an important toolset in remote sensing applications to enhance such spectral properties.

However, distinguishing canopy normal variations, drought and biotic stress requires these specific spectral data with high degrees of temporal and spatial resolution [2]. To meet such demands, PANTHEON has adopted the following cameras for UAV:

- *Sony© α5100* RGB camera
  - 3 broad visible red, green, blue bands
  - Focal Distance: 16-50 mm (35mm lens)
  - Pixel size: 3.92 µm
  - Width: 6000 pixels (23.5 mm)
  - Height: 4000 pixels (15.6 mm)

The high resolution of *Sony© α5100* enables the collection of surface details of the orchards. When overlaid with the multispectral and thermal sensor, the lower resolution of the multispectral and thermal cameras can be enriched by the fine resolution of RGB camera via spatial correspondence. For example in Figure 1, at a flight altitude of 20m, *α5100* still achieves a resolution of 0.5cm with significant details.

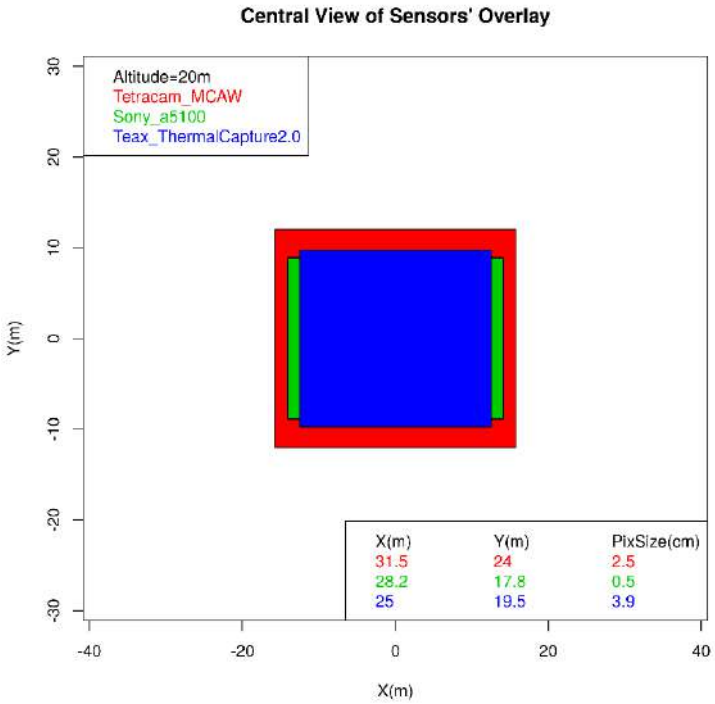


Figure 1: Simulation for three cameras image plane stack of a simultaneous shot

- *Tetracam© MCAW* multispectral band sensor.
  - 6 narrow bands 531nm, 550nm, 570nm 680nm, 720nm, 900nm
  - Focal Distance: 9.6 mm
  - Pixel size: 4.8 µm
  - Width: 1280 pixels (6.14 mm)
  - Height: 1024 pixels (4.92 mm)

The multispectral sensor *Tetracam© MCAW* provides in-situ spectral information specifically regarding to its photosynthesis activity in narrow bands. Typically, a band's spectral characteristics are determined by a spectral filter that has high transmissivity exclusively in desired wavelength range. Nearby narrow bands are capable to describe slight variations of the spectral features, but might be affected by a poor signal to noise ratio, since the energy received at the sensor depends on the bandwidth. Thus, high quality hyperspectral sensors are still very expensive.

In this case, PANTHEON favors *MCAW* for it allows users to customize transmissive filters. The filter configuration and the resulting spectral characteristics are organized in Table 1.

Illustrated by Figure 2, three regions are of major interests in a typical vegetation reflectance spectra: the green peak, the red edge, and the near- infrared (NIR) plateau. Important vegetation index such as normalized difference vegetation index (NDVI) [3] takes advantage of a vegetation's absorption in red energy and multiplicative NIR energy reflection due to its photosynthetic activities and physical structures to enhance such information by Equation (1). Another example is photochemical reflectance index (PRI) [4] that requires narrow band information as shown in Equation (2). The MCAW bands narrowly stand at two steepest but hard-to-capture locations near to the green peak, one at the top of green peak, one at the starting point before red-edge and one at the steepest slope red-edge, and one at NIR plateau, offering a great detail in fine spectral resolution.

Table 1 Spcetral characteristics of *MCAW*

Band	Center Wavelength (nm)	Bandwidth FWHM (nm)
Green	530.7	3
Green-Yellow	550.0	10
Yellow-Green	570.0	10
Red	680.0	10
Red-Edge	720.0	10
Near-infrared	900.0	10

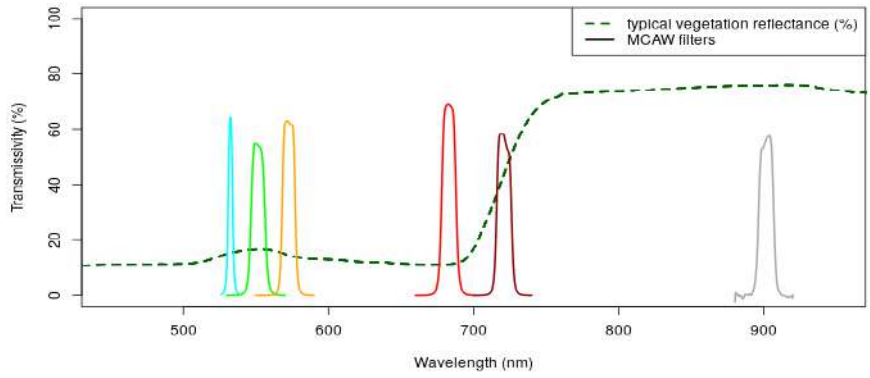


Figure 2 Filter transmissivity of *MCAW* along with a typical vegetation spectral reflectance

$$NDVI = \frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}} \quad (1)$$

$$PRI = \frac{p_{531} - p_{570}}{p_{531} + p_{570}} \quad (2)$$

- *TeAx© ThermalCapture 2.0* thermal sensor
  - 1 broad thermal band 7.5-13.5 µm
  - Focal Distance: 19 mm
  - Pixel size: 17 µm
  - Width: 640 pixels (10.88 mm)
  - Height: 512 pixels (8.704 mm)

The thermal sensor is a key asset for water stress and disease detection in this project based on the temperature symptoms when vegetation stomata reacts to water stress and disease. Water deficit closes plants stomata to reduce water loss from evapotranpiration. As a result, stressed plants are heated up slightly compared with well watered plants. This can be quantified by the crop water stress index (CWSI) [5] as described by Equation (3), where T_c is the plant surface temperature, T_{dry} is stressed plant temperature, T_{wet} is well watered plant temperature. This index might also be sensitive toward plant diseases due to affected stomatal conductance.

$$CWSI = \frac{T_c - T_{dry}}{T_{wet} - T_{dry}} \quad (3)$$

But accurate surface temperature is difficult to capture due to the changing ambient air. Therefore, a ground standard is offered by *Vaisala© WXT530* weather transmitter, which is deployed on the field to measure ambient temperature and irradiation, along with necessary humidity and precipitation information that is important for crop water dynamics.

Finally, a soil moisture monitoring network (*CAIPOS© CaipoRain*) is deployed in the field for crop water budget analysis.

The three cameras are mounted on a *DJI© Ronin-Mx* dual gimbal that can resist high G-forces by motors and inertial measurement unit (IMU). By triggering the cameras at once, multi-band images are captured simultaneously. The combined usage of these three cameras is the fundamental challenge of the aerial image processing pipeline. The collected images from them in the campaign are the base data for the experiments performed by WP4 and WP5.

3 Image pre-processing

For a camera operating in the field, three main types of errors are generated before a ray signal intensity being converted to a digital depth as illustrated by Figure 3 on the left. The first is an environmental attenuation such as aerosol absorption and scattering; the second is an optical transmission loss; and the final error is generated during the digital conversion.

Besides the signal intensity disruption, there are also ray geometric error as illustrated by Figure 3 on the right. The first is an image plane misalignment caused by the different camera positions on the gimbal. The second is ray deviation from rectilinear projection caused by optic geometry known as lens distortion.

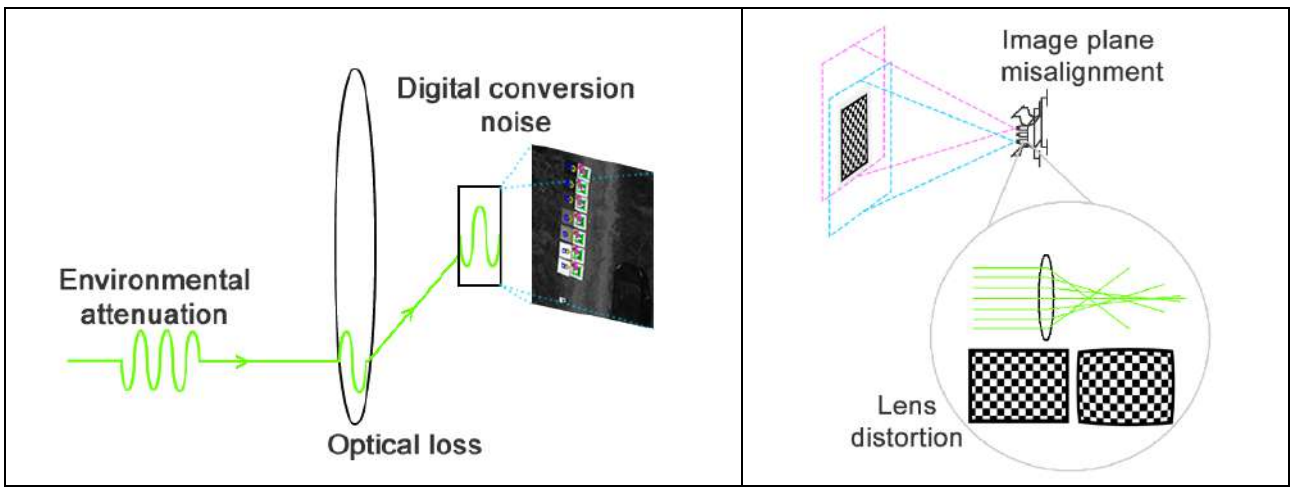


Figure 3 Ray intensity (left) and geometric error (right)

To create a well aligned multi-band image plane from all cameras without these errors, correction parameters can be derived from calibration images. These parameters can be divided into two main categories: 1) the invariant parameters that can be measured in laboratory, which are the optical transmission parameters, optical and gimbal geometric parameters; and 2) the field variant corrector parameters against environmental attenuation and digital noise.

Since several cameras are mounted on one platform, to organize all the shots from different cameras at one location, a virtual camera concept is introduced. Taking form as an "object" in Python language, a virtual camera object carries the necessary attributes and functions for each camera's calibration and image correction. When activated for calibration, it can derive and store the correction parameters by taking the calibration images as input. When activated for correction, it retrieves the correction parameters from database and correct the flight images.

As such, the whole pre-processing phase requires the following three steps:

- Laboratory calibration image retrieval at the beginning of the campaign
- Field calibration image retrieval at each trial data acquisition
- Implement calibration and correction by activating virtual camera object

In addition, a manual has been provided for calibration images retrieval in organized instructions.

3.1 File conversion

The image data type preferred in the whole processing phase is TIFF. *MCAW* already provides a multi-frame TIFF choice for image acquisition; *ThermalCapture2.0* is in multi temporal frame TMC format that can be converted by *TeAx's* data conversion library in C++ language; RAW file provided by *α5100* requires Python packages rawpy [6] and rawkit [7] to achieve that.

3.2 Laboratory calibration

Laboratory calibration is dedicated for the invariant optical and gimbal parameters that will not change during the campaign unless a physical micro deform occurs on the hardware. The two calibrations are:

- Vignetting calibration
- Chessboard calibration

3.2.1 Vignetting calibration and correction

The vignetting effect is a brightness reduction of an image towards periphery caused by optical transmission problems such as slight mechanical blocking or multiple lenses [8], etc. An image of a uniformly illuminated target without such effect is expected to have uniform pixel values with no systematic radiometric gradients.

A feasible approach to analyze vignetting effect is to capture a homogeneously illuminated white reference panel that completely covers a camera's field of view (FOV) several times. After noise removal mentioned in section 3.3.1, the average of them is a vignetting baseline image *A* that represents such effect.

Here, two variants of correction can be used, the first is to use the mean value of *A* to scale itself into a compensator such as:

$$\widetilde{I_{i,j}} = \frac{I_{i,j} \cdot \bar{A}}{A_{i,j}} \tag{4}$$

where $\widetilde{I_{i,j}}$ is the corrected image pixel value, $I_{i,j}$ is the original image pixel, $\bar{A}$ is the average value of vignetting baseline image *A*, $A_{i,j}$ is the pixel value of *A*.

The second method fits a two dimensional polynomial of degree four to the pixels randomly selected from $V = \bar{A}/A$, using the function *sklearn.processing.PolynomialFeatures* of Python package Scikit-learn [9]. The resulting polynomial factors can be used to remodel the vignetting image *V*. An image can be corrected by Equation (5), where $\widetilde{I_{i,j}}$ is the corrected image pixel value, $I_{i,j}$ is the original image pixel, $V_{i,j}$ is the pixel value of *V*.

$$\widetilde{I_{i,j}} = \frac{I_{i,j}}{V_{i,j}} \tag{5}$$

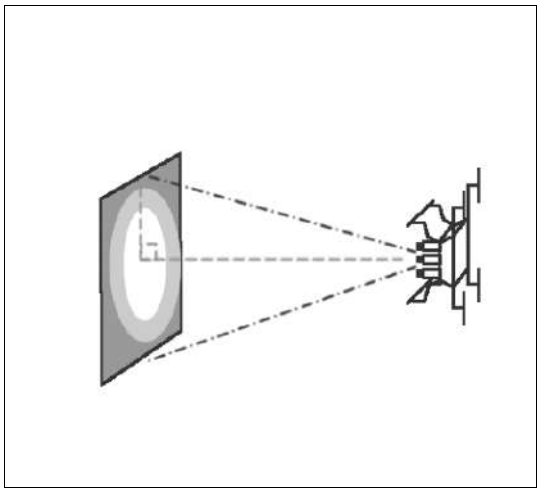


Figure 4 Illustration of laboratory vignetting calibration

To take vignetting calibration images in laboratory conditions, UAV mounted cameras should be positioned against a homogeneously illuminated flat white reference illustrated by Figure 4. Each camera's FOV should be completely covered by it. To ensure a homogeneous illumination, the white reference is placed in front of a window in laboratory, enforcing the indirect illumination only.

3.2.2 Chessboard calibration

The lens distortion problem is caused by the optical geometry, whose parameters including its focal length, skew coefficient, principal point, and lens distortion parameters known as intrinsic features. These features are uniquely possessed by a camera after its manufacture. For *α5000* and *ThermalCapture2.0*, there is only one lens. But for *MCAW*, the six bands are essentially six different cameras.

All cameras' different gimbal positions also cause significant image plane misalignment, which is a 3D geometric relation among all the cameras concerning each camera's pose.

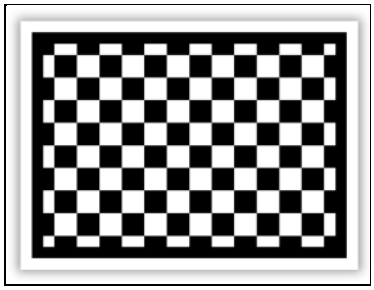


Figure 5 Chessboard marker

By capturing objects with known world coordinates, the 3D-2D geometric projection of the cameras can be reconstructed. Therefore, a chessboard pattern marker has been designed as Figure 5, whose intersection points in the 2D images can be detected automatically by *OpenCV* [10], the details of which can be found in *5.3.4.1 of D.4.1 (pp19)*. Since these intersection points have been defined with true 3D world coordinates, a 2D-3D geometric equation can be established similar to the content described in 4.2, Equation (5), pp17-18. Given a sufficient number of such equations, these parameters can be overdetermined [11].

As a result, the intrinsic features and the sensor alignment can be calibrated simultaneously. By providing the pose of an arbitrary point fixed within the gimbal, the UAV sensor pose can be calibrated as well. In such context, the sensor alignment provides either the roto-translation of each sensor in relation to a master band or the fixed point within the gimbal.

The intrinsic calibration and sensor alignment has been implemented in Python. After an automated detection of the chessboard markers, the camera calibration is performed by the *OpenCV's calibrateCamera* function, while the subsequent camera alignment is performed by the *stereoCalibrate* function with fixed camera parameters. The resulting calibration parameters—intrinsic matrix, lens distortion parameters and camera alignment matrices—are stored in the database in JSON format.

Since these parameters are used in *MetaShape's* bundle adjustment, no correction is involved here.

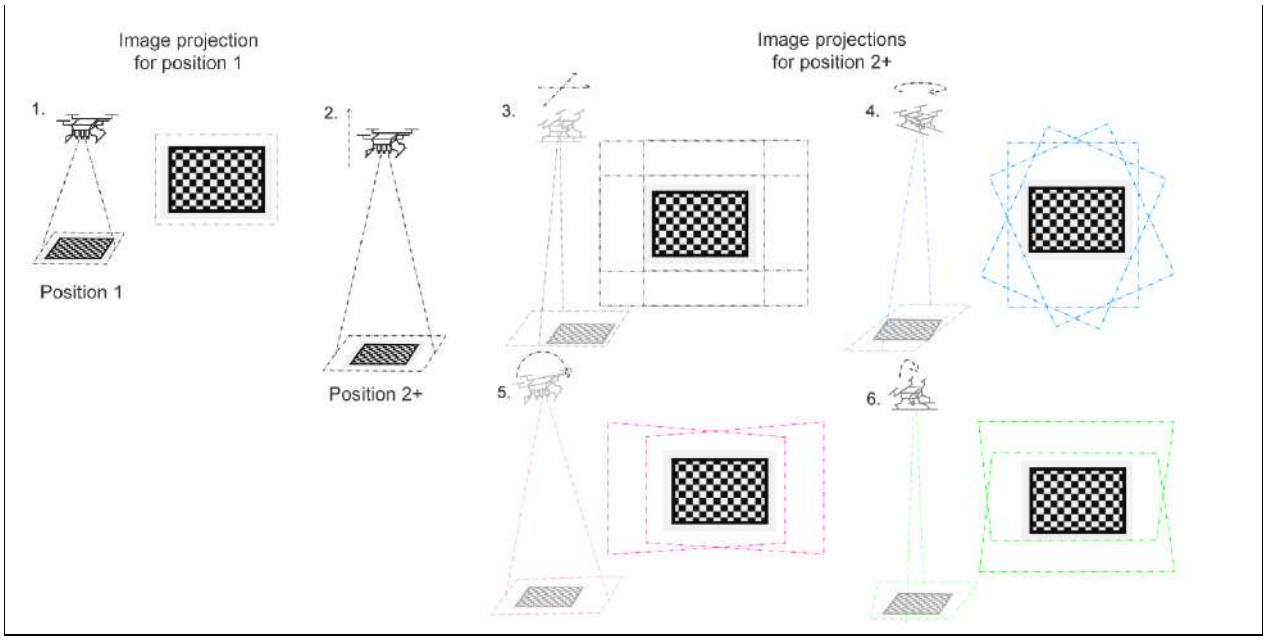


Figure 6 Laboratory chessboard calibration poses

To take chessboard images in laboratory conditions, the UAV mounted cameras are positioned over the chessboard in different positions and angles such as illustrated by Figure 6. Principally more variant positions will conclude better calibration results.

The suggest poses are as following:

- 1. Starting at Position 1, the chessboard lies as flatly and large as possible in each camera's FOV (estimated 50cm).
- 2. After that, the UAV increases altitude to Position 2+ (let the chessboard cover around 1/3 of each FOV) to take 1 image.

- 3. Then it horizontally translates to four directions (around 5cm) to take 4 images.
- 4. Moving back to original pose in Position 2+, it yaws 45°, 90°, and 135° for another 3 images.
- 5. Yawing back to original pose in Position 2+, the UAV rolls ±15° separately to take another 2 images.
- 6. Rolling back to original pose in Position 2+, the UAV pitches ±15° separately to take another 2 images.

3.3 Field calibration

Field calibration is dedicated for the variant parameters that will always change during the campaign according to the in-situ senario. The two calibrations are:

- Noise calibration
- Radiometric calibration

3.3.1 Noise calibration and correction

During acquisition, image noise is generated due to a variety of reasons ranging from the electronic circuit, digital converter and quantization noises, etc. It is assumed that the raw depth information of an image digital number (DN) is $DN_{raw}=DN_{rad}+DN_n$, where DN_{rad} is the meaningful radiometric information and DN_n the unwanted noises. The noise itself is composed of systematic noise that can be modelled by certain practical approaches and random noise that is difficult to quantify [12].

The exposure time and temperature have a major effect on the noise generation. By physically obscuring the cameras from the radiometric part, the “pure noise” itself can be observed from the dark current images for the specific exposure time at certain temperature. A detailed analysis of noise can be derived by computing its pixel-wise distribution from a large quantity of dark current images.

To take dark currents within field feasibility, the cameras should be covered by their manufactured lens capsules, taking 5 dark currents before the flight when the camera has already been powered on 3 minutes at least and another 5 more immediately after landing. The median of these images DN_{sn} is the systematic noise removal mask for the camera on that trial, and the variance of these images is the indicator for random noise fraction. To analyze the noise during laboratory calibration, 5 dark currents are taken after camera being powered on at least 3 minutes.

3.3.2 Radiometric calibration and correction

For a multispectral sensor, each pixel depth measures the amount of reflected radiance from the target in DN units. To study the true nature of the target free from atmospheric attenuations, calibrating DN to reflectance is a usual step. The most popular in-situ correction method is empirical line correction (ELC) [13], which requires a set of reference targets with known reflectance to be calibrated against their field measured DN via a linear regression model.

The project utilizes a set of seven field aluminium reference panels whose reflectance are blackroom measured. Their field DNs of representative pixels are required to be extracted from aerial images and correlated to the blackroom reflectance values. To identify the reference panels automatically and unambiguously in the field, a set of markers with unique IDs have been designed as explained in *4 of D.4.1 (pp12)*. When correctly placed along with the reference panels, the ID marker can be identified from an aerial image and used to extract reference panel’s field DNs by a Python defined marker detector. The size of the reference panels and markers is 75cm x 75cm, ensuring a robust automated marker detection and DN retrieval at an altitude up to 50m.

To determine the best nadir image for calibration, the minimum distance RMSE between the image centre and each marker's central position is calculated from marker-detected images. Overexposed reference panels are excluded from the calibration, since they would distort the regression line, this is automatically achieved by removing the DN that is close to the upper value range of the image.

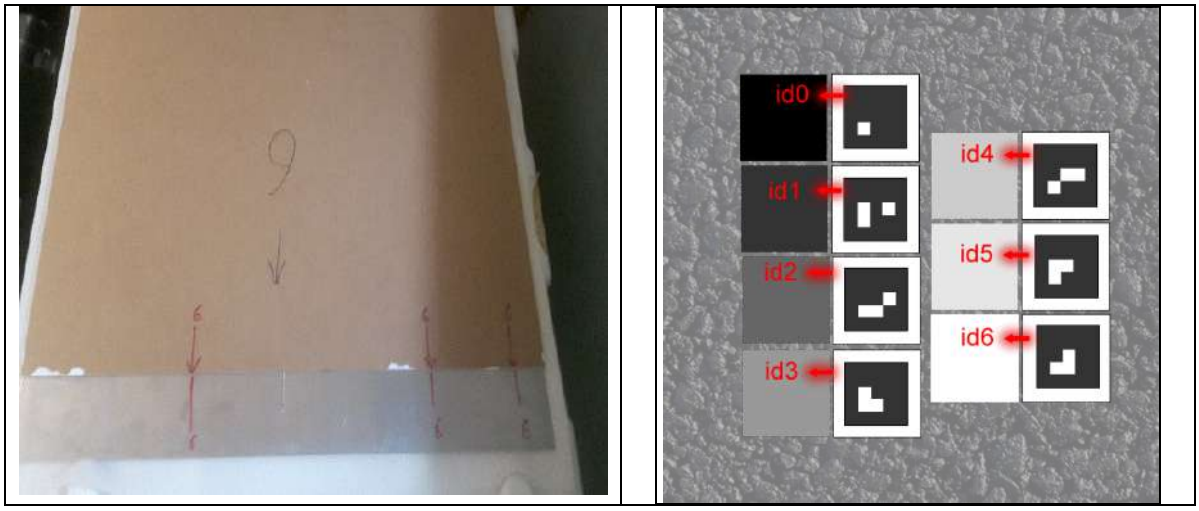


Figure 7 Reference panel and ID markers

To simplify the pairing of the reference panels and ID markers, their rear sides are labelled with the ID and three red lines in a unique pattern (Figure 7 left). To minimize the bidirectional effects and interference from background, a close formation of the reference panels and ID markers should be formed on a dark ground (i.e. asphalt) such as illustrated on the right of Figure 7.

The radiometric correction for thermal camera is more challenging than optical ones. Due to effects such as the inhomogeneous emissivity of target objects, a variety of noise sources like changing environmental and sensor temperature, accurate measurement of surface temperature is not feasible. To calculate the crop water stress index (CWSI) as an early indicator for water stress, supporting measurements such as air temperature, humidity, pressure and solar radiation, are collected by the ground station for thermal image for respective CWSI calibration.

3.4 Virtual camera object

The virtual cameras have three main functionalities:

- Implementation of camera calibration by loading calibration images
- Providing the calibration parameters to the database for reuse
- Performing the image correction by using the calibration parameters

These functionalities are organized in a structure illustrated by Figure 8. The basic workflow is to load the calibration images to the camera object to carry out calibration, then derive and store object parameters, and finally carry out the correction functions.

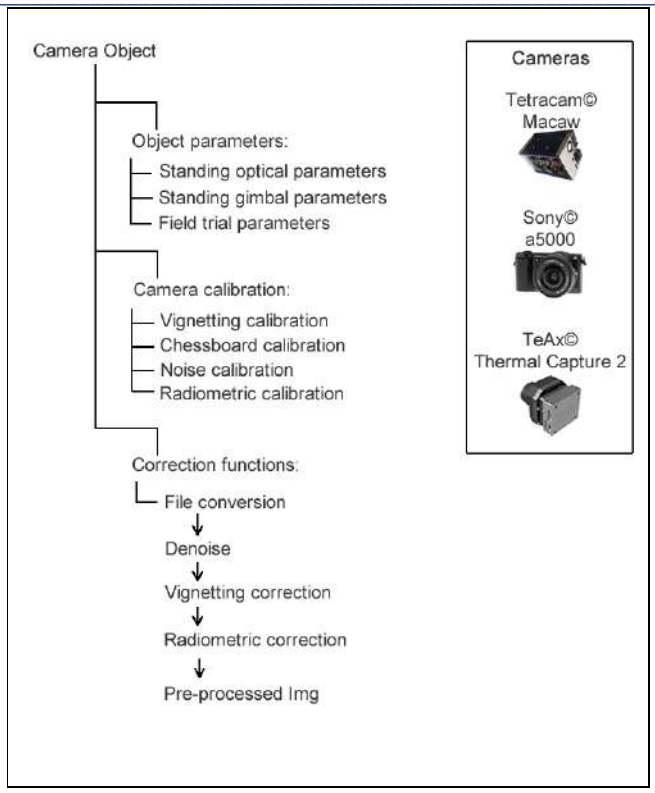


Figure 8 Virtual camera object structure for UAV

3.4.1 Implementation of the camera calibration

To execute both laboratory and field camera calibrations, the images from laboratory and field calibrations are loaded into the objects to derive correction parameters as described in the former sections. The calibration results are stored by SCADA for later steps. In principal, the laboratory calibration images are run only once at the beginning of the campaign unless another laboratory calibration is carried out; the field calibration need to be performed at each trial flight. Therefore, the frequency of a virtual camera object activation is per flight trial.

3.4.2 Object parameters

In the virtual camera object, the correction parameters from the calibration can be divided into mainly three groups, the standing optical and gimbal parameters and the field trial parameters.

The standing optical parameters include:

- the intrinsic features of camera, focal length f_x and f_y, principal points O_x and O_y, distortion parameters k_1, k_2, k_3, p_1, p_2
- the optical transmission parameter, vignetting average image A and polynomial parameters of vigenetting model V

The standing gimbal parameters are 3x4 band alignment matrix (rotation and translation offsets) for each band.

The field trial parameters include:

- median dark current image DN_{sn} (also for laboratory calibration)
- ELC linear coefficients for multispectral sensor

- and ground measurements (ambient temperature, radiation, precipitation, humidity and soil moisture) for thermal sensor

3.4.3 Implementation of the image pre-processing

The virtual camera object performs the correction as defined in the previous sections in Python by using these parameters. For signal intensity recovery, the general order is the backward process of Figure 2 on the left (pp9). Namely first denoising, then optical compensation, and finally radiometric correction.

Since the intrinsic and alignment matrix will be used in bundle adjustment, the intensity corrected images are not further distortion and band-align corrected, rather carry these parameters into the orthophoto computation.

4 Orthophoto mosaic computation

To orthogonally and accurately project the image pixels on a geographic raster plane, it is necessary to build a geometric relation between the 3D world structure and 2D photos. The whole process is achieved by utilizing *Agisoft© Metashape Linux* API in this project. The major processing steps of *Metashape* are following[14]:

- loading the photos into Metashape
- aligning photos (bundle-block adjustment to reconstruct the camera pose)
- building a dense point cloud
- building a digital surface model (DSM)
- building and exporting orthophoto mosaic

As illustrated by Figure 9, all the intermediate image processing results from an operation session are saved under a *"Project File"*. The basic object unit of each "*Project File"* is called a *"Chunk"*, which contains the images to be processed (camera pose and their corresponding photos), as well as processed objects like *"Tie Points"* (aligned feature points during photo-alignment), a *"DEM"* (the digital surface model according to user design) and the final result *"Orthomosaic"*.

The orthophoto mosaic computation module is composed of *Metashape's* Python commands. The command procedure is organized according to its normal standard as mentioned above. The module can be executed by the *Linux* system in headless mode.

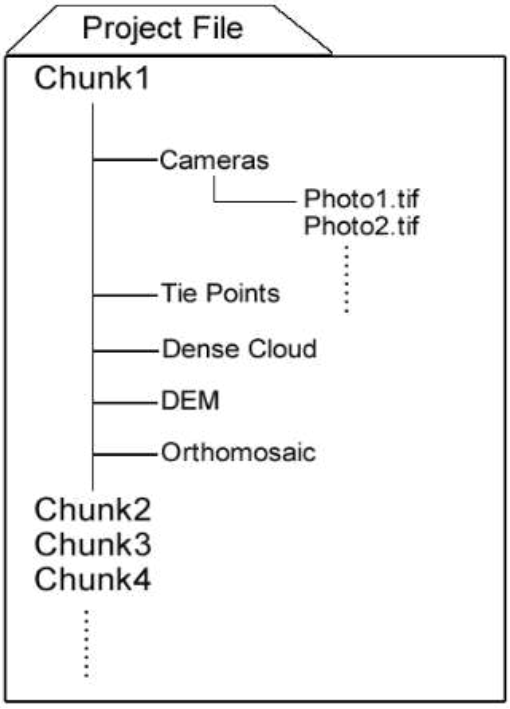


Figure 9 Visualised object hierarchy in Metashape

According to the current design, a project file contains only one chunk, which is composed of all the images from one flight. Several chunks can be combined to create a much larger extent if necessary.

4.1 Loading images into Metashape

For an aerial map of *k* bands, the physical addresses of the images at each camera position are loaded in a *"Camera"* vector such as

$$Camera_1 = [\text{path/image}_1.\text{tif}, \text{path/image}_2.\text{tif}, \quad \ldots \quad , \text{path/image}_k.\text{tif}] \tag{6}$$

For one flight that has triggered *n* times of shots, vectors $\text{Camera}_{1,2\ldots n}$ are appended into the *Chunk.Cameras* object via method *"MultiplaneLayout"*. The most monochrome sensitive band is designated as the master band.

After camera vector appendix, a list of all camera's pose (GPS, altitude and orientation) is provided to the chunk as initial poses. The physical image addresses and the required metadata will be provided by SCADA system such as described in *3.1 of D.4.1 (pp10)*.

4.2 Photo alignment

The *"Photo Alignment"* is the initial step to construct 2D-3D connections from the images in a rough but robust and effective way. Generally, it consists two steps: 2D robust feature points matching and their 3D reconstruction.

It starts by the feature detection through all the images via gradient sum techniques similar to Scale-Invariant Feature Transform (SIFT) [15] algorithms to recognize, descript and match points. Recognizable scale invariant features like "corners" and "edges" [16,17] are designated as "key points", and the key points that can confide in a robust match between image pairs are called "tie points".

Unlike the known 3D coordinates of chessboard intersection introduced by section 3.2.2, the only position well known here is a tie point's 2D image coordinates through feature detection and its match in other

images. By establishing a geometric equation between a point's 2D coordinate and 3D true coordinate, the equation parameters can be overdetermined when a sufficient number of points are detected from image pairs of different viewing poses. When initial values such as camera pose and lens intrinsic features are given, the solution becomes an iterative optimization adjustment.

The relation between a tie point's homogeneous position vector in pixel coordinate *u'=(u,v,1)* and camera perspective coordinate *X'=(X,Y,Z)* is described by Equation (7):

$$Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} f+B_1 & B_2 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{pmatrix}}_{I}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{7}$$

The intrinsic matrix *I* contains the focal length of a camera *f*, an affinity corrector B_1, a skew factor B_2, the principal point coordinates O_x and O_y, while *Z* is the camera distance.

When the lens distortion parameters are further considered, then the camera distance *Z* on the left side of Equation (7) is moved to the right such as the second matrix multiplication in Equation (8), where the unit-less perspective coordinate vector $(X/Z, Y/Z, 1)^T$ is subject to a division Brown-Conrady distortion corrector function *d()* [18,19].

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f+B_1 & B_2 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{pmatrix} d\begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{pmatrix} = \begin{pmatrix} f+B_1 & B_2 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{pmatrix} d\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \begin{pmatrix} f+B_1 & B_2 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} \tag{8}$$

In Equation (8), $(x_u,y_u,)$ in the middle is the "undistorted" pixel location projected on 2D image coordinate from camera perspective coordinate that is computed by bundle adjustment, being the unit-less *X/Z* and *Y/Z*. (x_d, y_d) is the distorted pixel location in camera perspective.

The function *d()* describing the relation between $(x_u,y_u,)$ and (x_d, y_d) is expressed by Equation (9), where $r^2=x_u^2+y_u^2$; K_1, K_2, K_3, K_4 are the radial distortion coefficients; P_1, P_2, P_3, P_4 are the tangential distortion coefficients.

$$\begin{aligned} x_d &= x_u(1+K_1r^2+K_2r^4+K_3r^6+K_4r^8)+[P_1(r^2+2x_u^2)+2P_2x_uy_u](1+P_3r^2+P_4r^4) \\ y_d &= y_u(1+K_1r^2+K_2r^4+K_3r^6+K_4r^8)+[P_2(r^2+2y_u^2)+2P_1x_uy_u](1+P_3r^2+P_4r^4) \end{aligned} \tag{9}$$

A point in the world coordinate system expressed as homogeneous vector $X_0'=(X_0, Y_0, Z_0, 1)$ and its camera perspective coordinate *X'=(X, Y, Z)* is described by Equation (10):

$$X' = EX_0' \tag{10}$$

where *E* is the extrinsic matrix explained by Equation (11), consisting of rotation *R* and translation *t.*

$$E = (R|t) = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{pmatrix} \tag{11}$$

The details of *Metashape's* bundle adjustment model is attached in the Appendix to be compared with *OpenCV* model. In uncalibrated camera situation, only *(u,v)* is known, all the others are to be derived through overdetermination. But as introduced in 3.2.2, the intrinsic matrix in Equation (8) and distortion parameters in Equation (9) are known. The extrinsic matrix in Equation (11) is further adjusted by the band alignment matrix. This is achieved by defining the parameters in *chunk.sensors* object instances.

The computation is achieved by the API command *chunk.matchPhotos* with a series of function parameters, including the uplimit of key and tie point amount, the accuracy setting of pixels amount for feature detection, and image pair preselection rules. At this stage, the default maximum setting computes 40,000 keypoints and 10,000 tie points for each image. The default accuracy is *"high"*, working on the original size of an image. GPS dependent *"Pair preselection"* that fastens the image pair selection is naturally enabled.

The result of reconstructed tie points in world coordinates and along with the key points in a given projection system is called *"sparse point cloud"*.

4.3 Dense point cloud generation

Although given the sparse point cloud that describes the 3D scenario with the most obvious feature points, a denser 3D model is required to describe the relation between each image's fine elements (i.e. image pixels themselves) and the real world. To reconstruct such a *"dense point cloud"*, the so called densification step also requires two general steps: fine elements 2D matching and their 3D reconstruction.

With the confirmed camera poses from the last section, the corresponding finer elements are determined between image pairs in epipolar constrains by generating the so called *"depth maps"*. The involved geometry equations may lead to many possible solutions, therefore the neighborhood information of the elements are also included to determine the right matches. This is achieved by *chunk.buildDepthMaps*. The map is then immediately followed by the geometry reconstruction similar to the last section by command *chunk.buildDenseCloud* within an optimized structure such as patch multi-view stereo (PMVS) algorithm[20].

During depth map generation, two parameters are of importance. The first is *"quality"* that determines the upper limit of the fine element amount, with *"ultra high"* computing all the pixels from a source image. To balance efficiency and quality, the 50% pixels of one image *"high"* set is chosen. The second is *"filtering mode"* for noise removal. Image noises could be contributed by badly focused images or illumination variation. To the rule of thumb, the more canopy detail is needed, the less filtering should be. But under varying illuminations such as passing clouds, robust match may not be achieved during depth map generation. When such an occasion occurs, the strategy is to first ensure a densification as accurate as possible, where the *"aggressive"* mode is enabled to sort out most outliers.

4.4 Digital surface model generation

Given a dense or sparse point cloud, a Digital Surface Model (DSM) is interpolated by inverse distance weighting (IDW) [21] the elevation values among the points on the spatial area. The computation can be processed both on sparse and dense point cloud. The DSM's detail level from the former is lesser than the latter, but much faster if the dense point cloud generation is bypassed. The detail of DSM essentially affects the projection locations of all the pixels on the raster. This is achieved by command *chunk.buildDem*, where a source file either *"PointCloudData"* or *"DenseCloudData"* can be defined by user.

The pipeline at this stage includes the dense point computation as a default step in the chain due to the need for canopy detail analysis, and sets *"aggressive"* filtering as default, as suggested by *Metashape* user manual, due to the flat nature of the study area (Figure 23). Further adjustments might be made based on the validation experiment on the field.

4.5 Building and exporting orthophoto mosaics

Based on the DSM, the computation completes the process by image ortho rectification. By manipulating the extrinsic matrix of each image, each pixel is orthogonally projected on the DSM. The pixels covered by the higher pixels in the real world coordinate system are discarded. This is realized by *chunk.buildOrthomosaic* command in API, where the *"surface"* source is referred to the *"ElevationData"* (DSM) computed in the previous step. The blending of the overlapping pixels is set as default *"mosaic"*, which *divides the data into several frequency domains. The highest frequency component is blended along the seamline, where further component are less subject to the blending* [14].

To export the result, the command *chunk.exportOrthomosaic* lies in the final position of the module. The pixel size and projection system of the exported mosaic result is fully user-demand dependent.

Although prepared with real-time kinematic (RTK) GPS system for the UAV, the possible delays of the signals are the often occasions, causing matching errors in the final mosaics. To tackle this possible challenge, the pipeline might include geo-reference markers in later steps based on the first test flight results. This procedure requires a deployment of three or more geo-markers on the corners of a study field. The markers in the field should be measured with ground GPS. For automation, the markers should be permanently fixed in the field.

In this case, additional steps should be added into image pre-processing, where geo-reference markers search is implemented through all the master band images. The image GPS is compared with the list of ground measured GPS positions when a reference marker is found. The closest ground GPS is thus the markers' GPS. Paired with their pixel coordinates, these GPS – pixel coordinate pairs can form a geo-reference marker list for *Metashape*.

4.6 Orthophoto mosaic implementation

Currently, a Python module has been implemented to execute all the above steps in one chain. It is planned to split the function into specific functions in the next stage.

The pipeline requires four input variables to be defined and called by the SCADA system, being the physical address of one file that contains all the images of one flight, the physical address of a GPS list (such as a csv file), and the desired destinations for saving project files and the orthophoto mosaic result.

The processing pipeline has been implemented in a modular design to ensure a high level of flexibility. For each image pre-processing step an individual Python script will be called. The modules access the database directly to extract and store all relevant data. Since the *MetaShape*'s Python API is accessed by passing a Python script, the SCADA system is planned call *Metashape* in headless mode after system integration. For each day of flight an individual orthophoto mosaic will be generated.

5 Results and test experiment

5.1 Image pre-processing test

5.1.1 Laboratory calibration and correction test

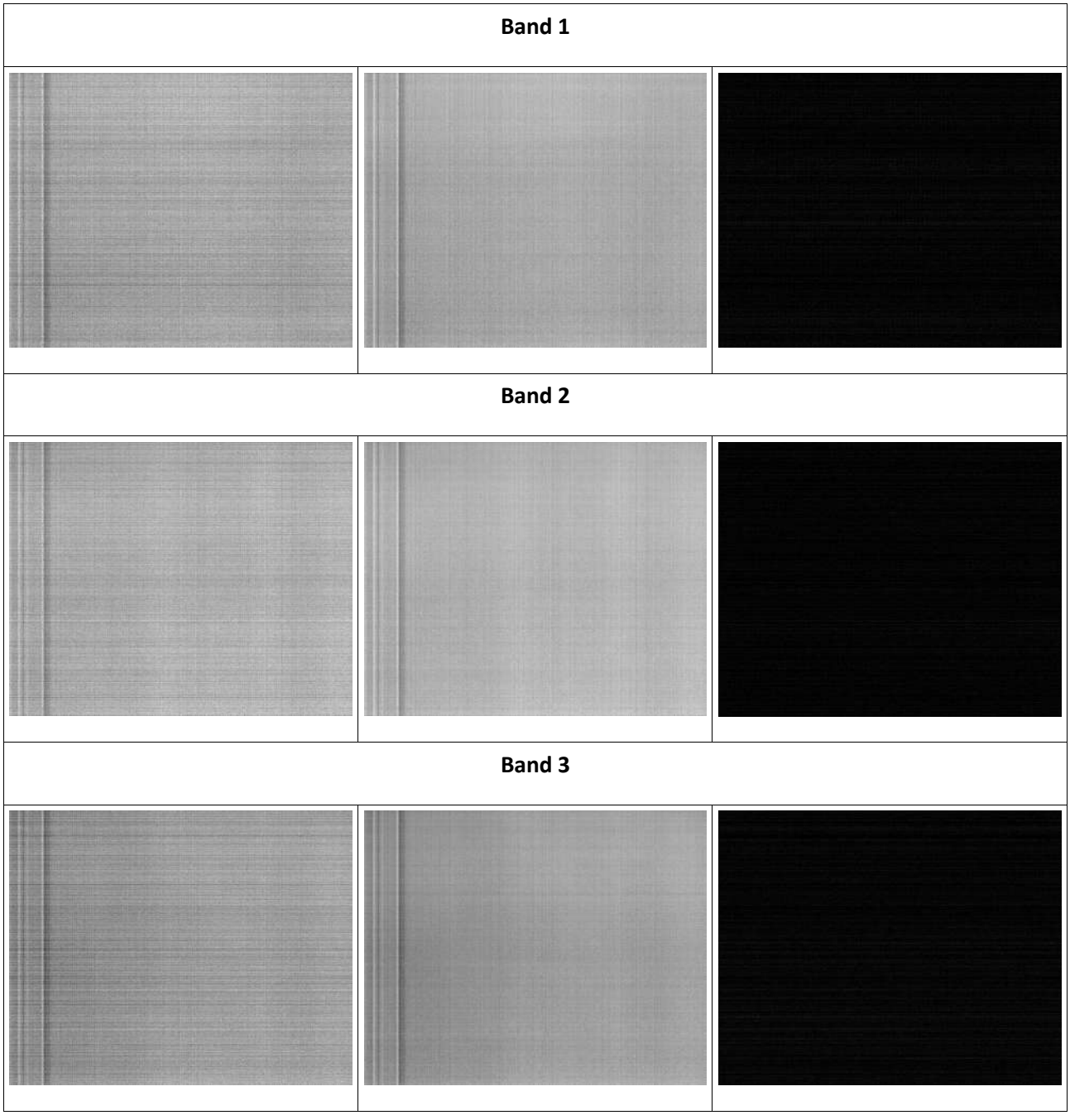
The laboratory calibration of UAV is largely identical to UGV, where *D.4.1* has tested and concluded a series of laboratory calibrations from data taken at the end of November 2018. These include the vignetting correction in *7.4.1 of D.4.1 (pp29)*, distortion calibration in *7.4.2 of D.4.1 (pp32)*, and sensor alignment in *7.6 of D.4.1 (pp35)*.

5.1.2 Field calibration test

Since the first phase of the project has no UAV flight plan, the noise calibration test presented here was only laboratory for *MCAW*; for field radiometric calibration, a configuration of *MicaSense© RedEdge* on a *DJI© Phantom4* was tested in Germany.

5.1.2.1 Noise calibration and correctiontest

The multispectral sensor *MCAW* is characterized by a systematic noise pattern (see Figure 10). To test the algorithms implemented for dark current correction, 30 MCAW images with covered lens were taken in the laboratory. The resulting median dark current images, as well as examples of corrected images are illustrated in Figure 10.



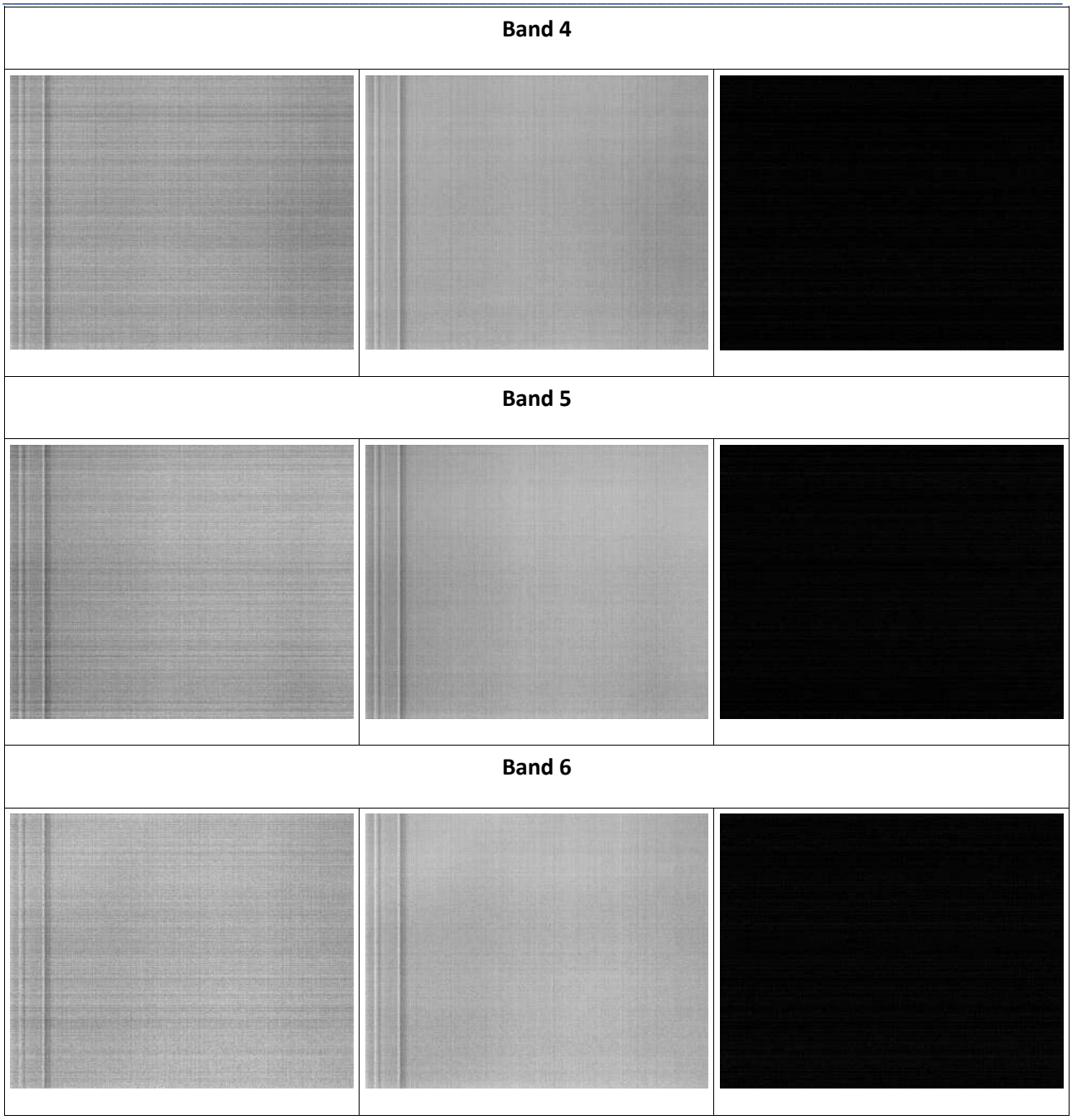


Figure 10 Illustration of the dark current correction for all six MCAW bands. Example image taken with covered lens before dark current correction (left) and after dark current correction (right) using the median dark current image (center). For illustration, the DN values are linearly stretched from 0 to 1000

Table 2 summarizes the results of the dark current correction. Since dark images without noise are expected to have a uniform value of zero, the standard deviation here is a good noise indicator. As expected, the standard deviation is decreased by dark current correction. Thus, the corrected images are characterized by random noise only. The explained variance corresponds to fraction of systematic noise DN_{sn} in relation to the total noise DN_n. Based on the given results, about 50% to 70% of the *MCAW* bands noise can be removed by dark current correction.

Table 2 Results of the dark current calibration

MCAW band	mean standard deviation before correction (DN)	mean standard deviation after correction (DN)	explained variance (%)
Band 1	112.6 ± 1.2	53.0 ± 7.3	56.1 ± 1.2
Band 2	101.6 ± 0.7	48.6 ± 6.1	68.9 ± 1.0
Band 3	115.2 ± 1.7	51.5 ± 8.5	53.6 ± 1.6
Band 4	99.7 ± 1.1	47.1 ± 7.0	71.5 ± 1.5
Band 5	111.0 ± 1.4	50.5 ± 7.1	57.7 ± 1.4
Band 6	115.8 ± 1.0	50.9 ± 7.8	± 0.1

5.1.2.2 Radiometric calibration and correction test

The multispectral sensor *RedEdge* images were tested with the automatic detection of ID marker, reference panel DN retrieval and ELC computation. The detection result is visualized as Figure 11. To illustrate the ELC coefficients computation, two bands' ELC regression lines are visualized in Figure 12 and 13. The high R^2 suggest a very good correlation between the DN and black room measured reflectance.

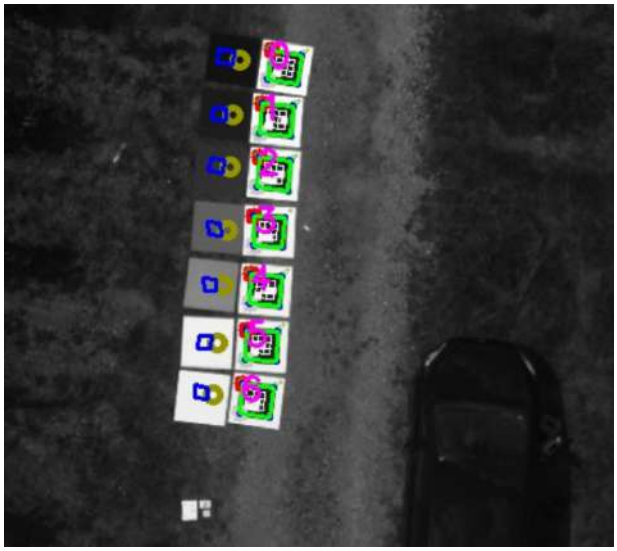


Figure 11 An aerial image augmented with the detected reference panels. The numbers (violet) indicate the identified marker ID, while the areas enclosed by blue lines are used for the extraction of the DNs.

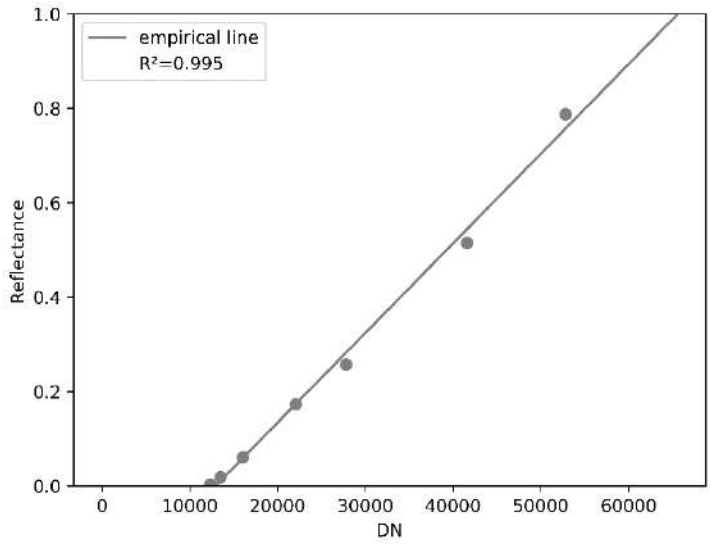


Figure 12 Empirical line on band4 Near-infrared (840nm), R^2=0.995

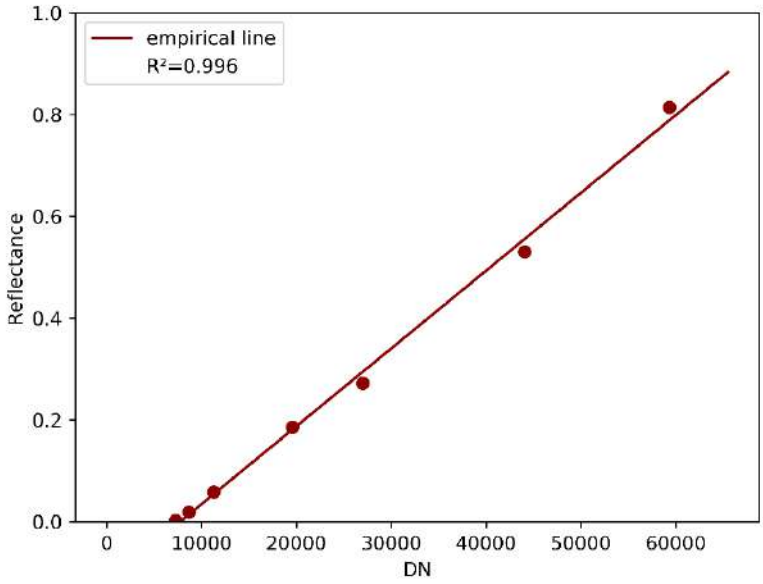


Figure 13 Empirical line on band5 red edge (717nm), R^2=0.996

5.2 Orthophoto mosaic computation test

The orthophoto mosaic results from three trial flights are presented here with NIR, red and green false color composition. The test side was a slope vineyard in Bernkastel-Kues, Germany. The three flights were conducted on 21 Jun good weather, 23 August change clouds and 12 Sep good weather in 2018. The general overview of the orthomosaic mosaic results are illustrated by Figure 14-16. These datasets represent three kinds of scenarios: well developed canopy structures under good weather condition, pruned canopies under bad weather condition, pruned canopies under good weather conditions.

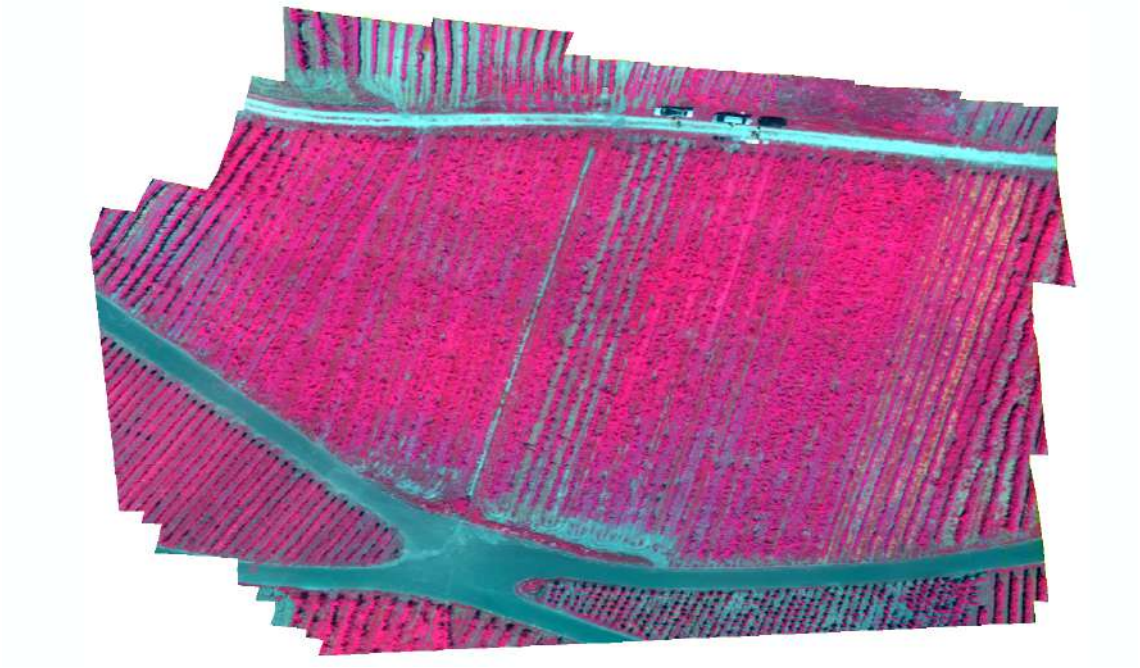


Figure 14 Orthophoto mosaic on 21 Jun (good weather, dense aggressive filtering)

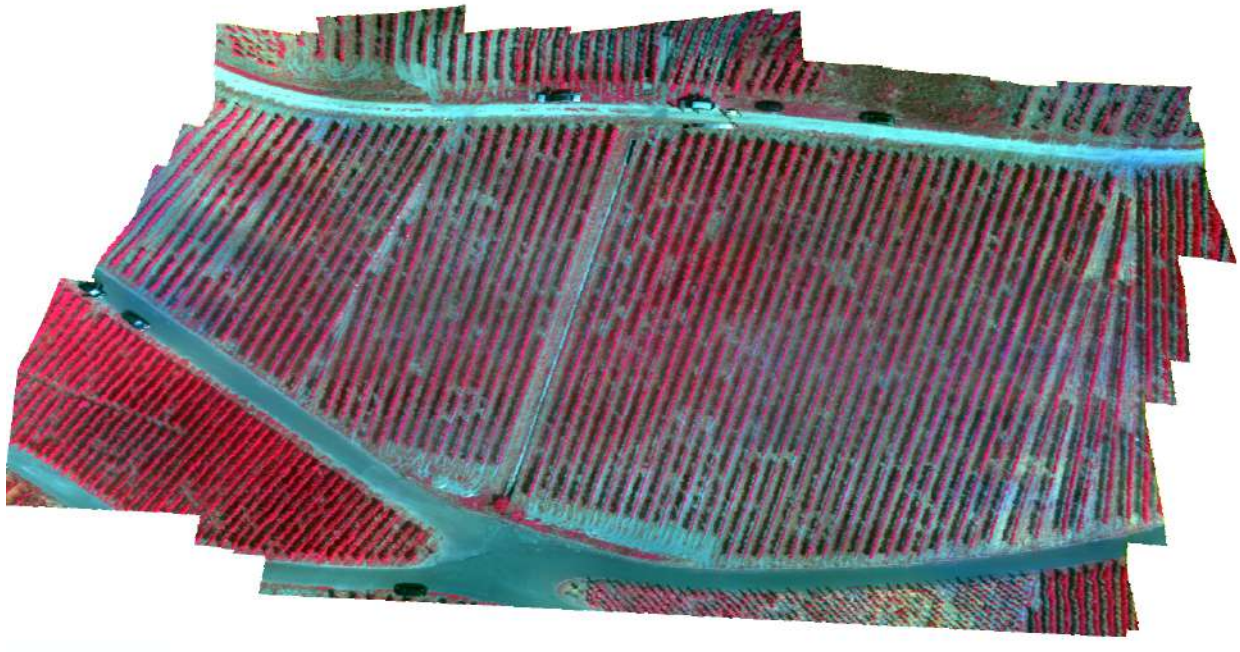


Figure 15 Orthophoto mosaic on 23 Aug (varying clouds, dense aggressive filtering)

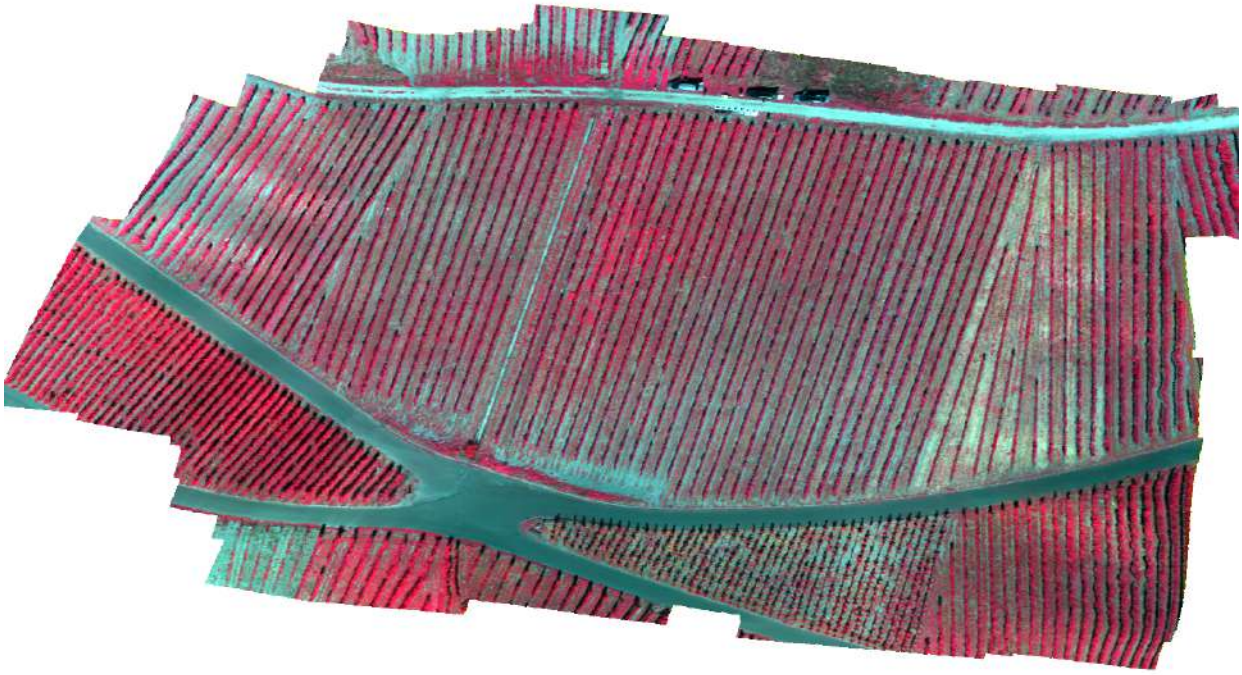


Figure 16 Orthophoto mosaic on 12 Sep (good weather, dense aggressive filtering)

Figure 17 and 18 illustrate the orthophoto mosaic results and their DSM on 21 Jun (good weather, rich canopy) and 23 Aug (bad weather, pruned braches). Intuitively the mosaic results do not seem to differ too much with different DSMs.

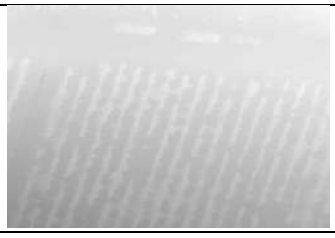
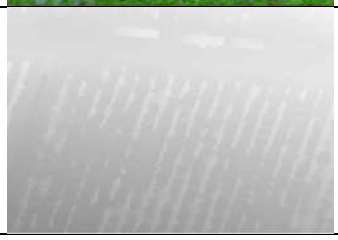
	Sparse point cloud	Dense point cloud no filter	Dense point cloud aggressive filter
Orthophoto mosaic result			
DSM			

Figure 17 Mosaic results and DSMs on 21 Jun 2018, good weather

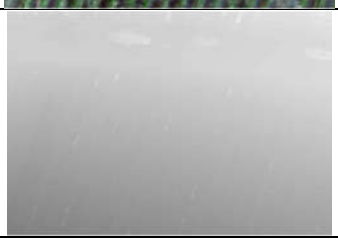
	Sparse point cloud	Dense point cloud no filter	Dense point cloud aggressive filter
Orthophoto mosaic result			
DSM			

Figure 18 Mosaic results and DSMs on 23 Aug 2018, bad weather

The spatial detail of DSM differential subset from Figure 19 to 22 yield more information on this matter. Figure 19 suggests that more than 20% of the pixels from a sparse and no filter dense DSM differential can differ larger than 0.2m, the maximum differential can go high up to 2m. Also, the surface details are less from the sparse DSM.

Figure 21 suggests that filtering mode can generate around 5% of the subset to differ larger than 0.2m. The differentials have a very narrow distribution around 0 as illustrated by Figure 22, meaning a not very significant difference distribution between the no and aggressive filtering in this subset.

The sparse point cloud stands for the most obvious features in the 3D world, while the dense no filter cloud is the most detailed one that might contain noises, and the dense aggressive filter cloud removed certain part of the outliers. Since a large spatial 3D validation is hard to check which is the best option here, a balanced choice among the methods should be tested with the real situation in the study field.

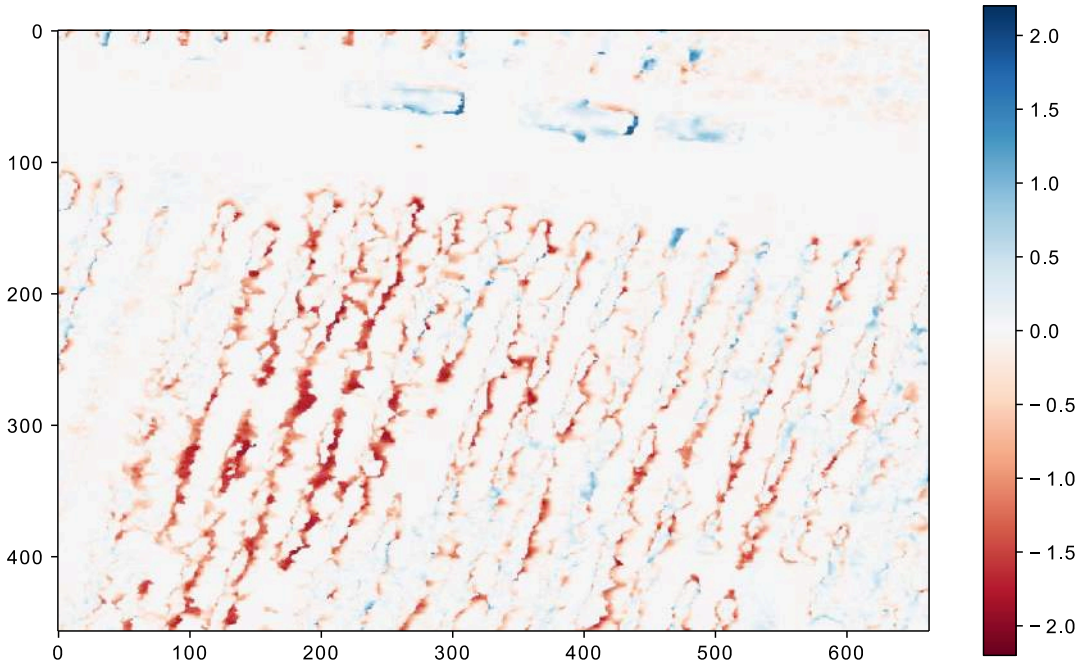


Figure 19 DSM differential (dense no filter - sparse) subset on 21 Jun 2018, good weather, subset RMSE 0.363m, 80.64% of the pixel absolute values are lower than 0.2m (white color). The widely distributed color pixels suggest that the sparse DSM and dense no filter DSM have a quite large spatial differences on the canopy structure detail.

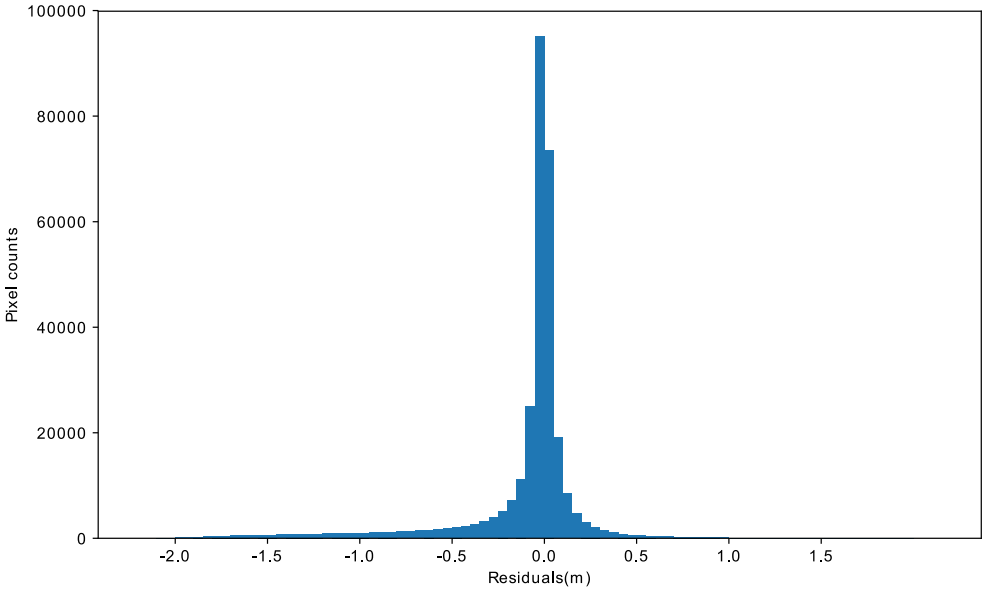


Figure 20 Pixel value histrogram corresponding to Figure 19

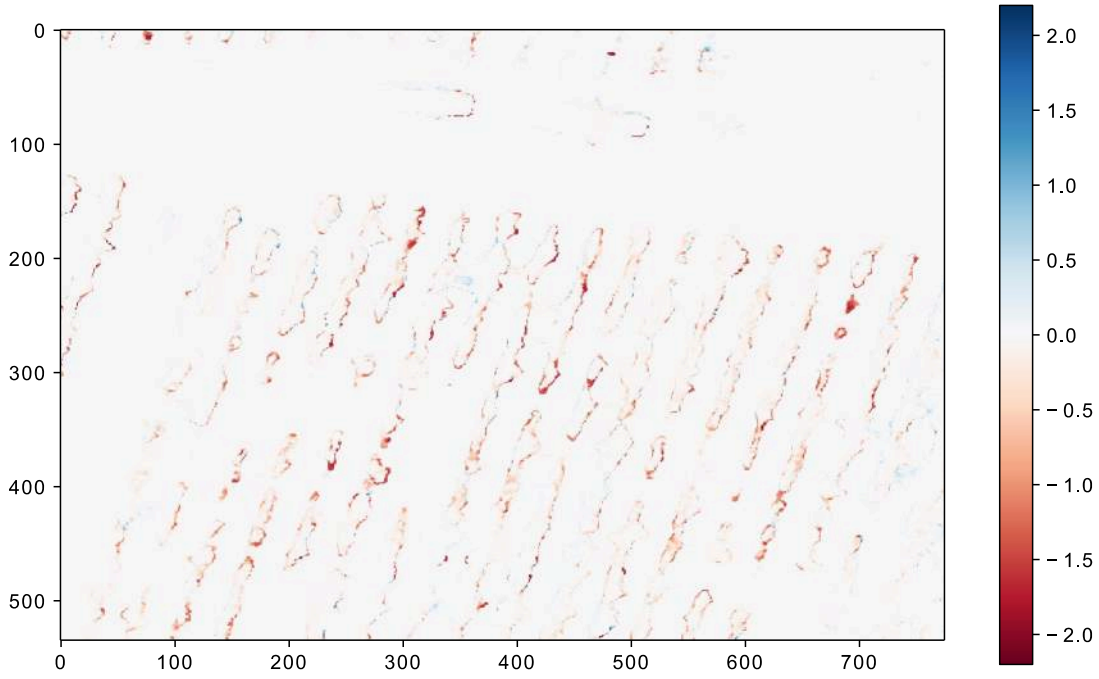


Figure 21 DSM differential (dense aggressive filter – no filter) subset on 21 Jun 2018, subset RMSE 0.176m, 95.51% of the pixel absolute values are lower than 0.2m (white color). The majority negative values surrounding the crops suggest that the aggressive filter removed these as outliers.

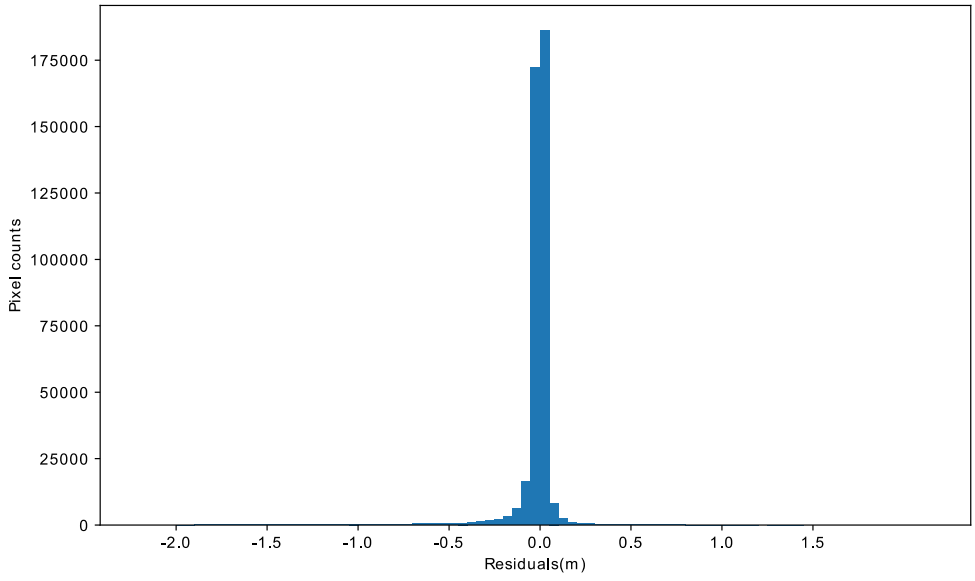


Figure 22 Pixel value histrogram corresponding to Figure 21

5.3 Upcoming field trials

To address PANTHEON's objectives, a set of field trials have been updated by the consortium as discussed in *D.2.3 (pp7)*. The UAV will collect the relevant remote sensing data as illustrated by Figure 23, the study area covers two field parts of different characteristics. The left field consists of old orchards with a survey area of approximately 0.36ha. The right field is a composite of young trees with a survey area of approximately 0.40ha. Next to the agrometeorological requirements, the distribution of the treatment groups has been designed for a possible full trial coverage within one or two UAV flights to minimize the expenses for data collection.

The reference panels are planned to be placed next to the road or between the fields. To protect them from dirt, moisture and dust, options are reviewed that they should be placed in flat plastic boxes and left in the field permanently. To validate the geometric accuracy of the orthophoto mosaics for each flight, it is planned to place markers with known GPS positions in the field. These markers also serve as a fallback in case of RTK malfunction.

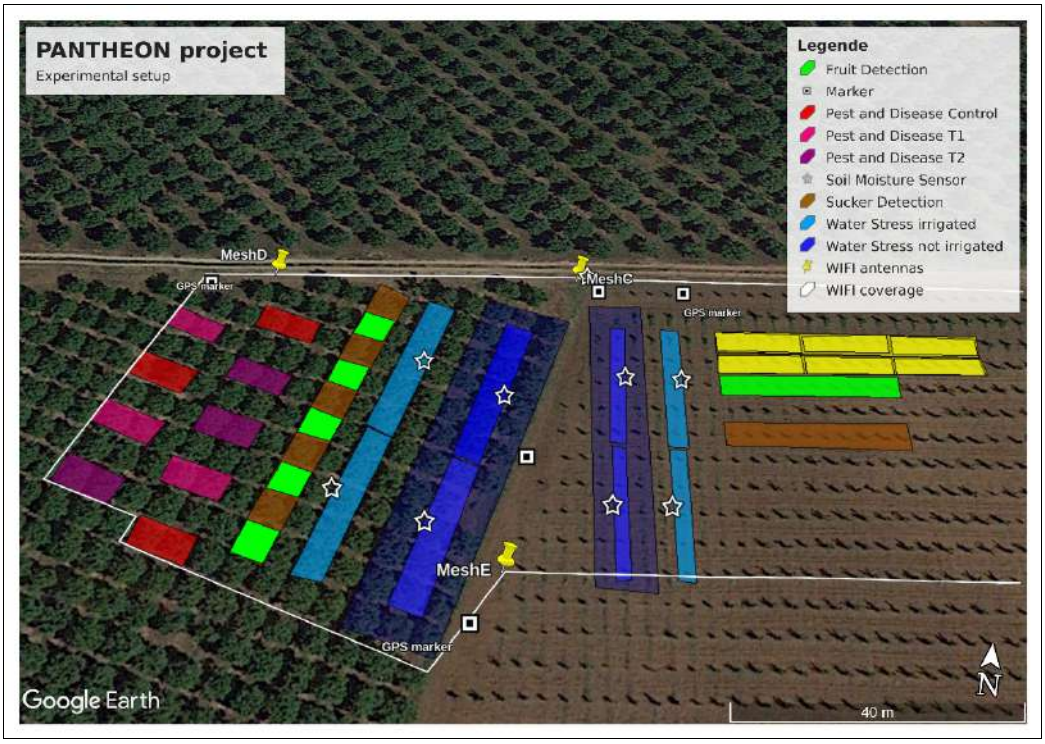


Figure 23 Upcoming study field

5.4 Challenges during the experimental setup

The previous test setup pushed some challenges for the development of the processing pipeline. The most important difference to the UAV used by PANTHEON is the combination of sensor systems. While during the test stage, only one multispectral camera was available for mosaic computation test.

Although the UAV could collect GPS information for each image, but was not capable to provide IMU and RTK information as intended for the SCADA system envisioned by PANTHEON. This lack of accurate information, in combination with a steep terrain and varying weather conditions hardened the selection of sufficient parameters for high quality mosaic generation.

To overcome the problems of the differing sensor systems, the pipeline has been module standardized. Namely, most of the code is used in both UAV and UGV pipelines, the shared calibration methods such as

vignetting correction or camera calibration have been designed to be sensor independent. To ensure a high level of modularization, the metadata processed in each image pre-processing step are stored in the database for later reuse, while each subsequent processing step is intended to extract the relevant metadata from the database.

Since PANTHEON's SCADA system will provide much more accurate UAV metadata, the processing pipeline is expected to be more robust and more accurate results for the hazelnut orchard.

6 Conclusions

6.1 Completed tasks

The priority of *Task 4.5 Aerial Image Processing Pipeline* is to deliver an automated image processing and orthophoto mosaic computation pipeline for *Task 4.6 Water Stress Measurement* and *Task 4.7 Pest and Disease Detection*.

Algorithms and protocols have been developed in close cooperation with *Task 4.2 Terrestrial Remote Sensing Pipeline* in terms of image pre-processing by *OpenCV* and orthophoto mosaic computation based on *Metashape*. Together, these two parts form an operating pipeline that has been preliminary tested by individually configured aerial robot available during the development stage.

6.2 Ongoing tasks

Since the first phase of the project was not intended with aerial measurement, the interaction between the processing steps was not directly tested on the dedicated cameras. Instead, the modules in the pipeline have been stand-alone validated with their functionalities. It is expected that the sensors will be calibrated once the UAV is working reliable and the first field experiments to be performed.

Based on the results of the first field tests, minor changes of the processing might be forwarded. In particular, the parameters used for orthohphoto mosaic generation need to be optimized. Thus, the results of the pipeline will be validated continuously. Since the server structure of the SCADA system is still under development, some modifications of the pipeline are also likely to be expected for the system integration, such as database connection protocols since only one local file database could be used during algorithm development. In addition, the automatic conversion of the *TeAx* files needs also to be integrated.

6.3 Guidelines for field experiments

Based on the test results and common practice, the UAV field campaigns are required to be performed under clear sky conditions. In particular, passing clouds during measurement need to be avoided to ensure a good quality of the orthophoto mosaics.

In addition, single flight mosaics are preferred towards multi-flight mosaics, since changing sun angles corrupt the quality of the empirical line correction.

To counter the airflow effects for thermal sensor, it is also suggested that a short ambient air temperature adaption pre-flight should be taken before the actual trial flight begins.

7 References

[1] Luhmann, T., Robson, S., Kyle, S., Boehm, J., "Close-Range Photogrammery and 3D Imaging 2nd Edition" (pp323), De Gruyter, (2014).

[2] Gerhards, M., Schlerf ,M., Rascher, U., Udelhoven, T., Juszczak, R., Alberti, G., Miglietta, F., Inoue, Y., "Analysis of Airborne Optical and Thermal Imagery for Detection of Water Stress Symptoms ", International Journal of Remote Sensing, *Remote Sens.*, 1139, *10*(7), (2018).

[3] Rouse, J.W.; Haas, R.H.; Deering, D.W.; Schell, J.A. "Monitoring the Vernal Advancements and Retro Gradation of Natural Vegetation", Taxas A & M University: College Station, TX, USA, (1974).

[4] Gamon, J.; Peñuelas, J.; Field, C. "A narrow-waveband spectral index that tracks diurnal changes in photosynthetic efficiency", Remote Sens. Environ. 41, 35–44, (1992).

[5] Jackson, R.D.; Reginato, R.J., Idso, S.B. "Wheat canopy temperature: A practical tool for evaluating water requirements". Water Resour. Res., 13, 651–656, (1977).

[6] Riechert M., letmaik/rawpy, (2018).

[7] Paul C., Whited S., rawkit 0.6.0 documentation, (2018).

[8] Kordecki, A., Palus H., Bal, A., "Practical vignetting correction method for digital camera with measurement of surface luminance distribution", Signal, Image and Video Processing, Bd. 10, pp.1417-1424, 7 (2016).

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, Bd. 12, pp. 2825-2830, (2011).

[10] Bradski, G., "The OpenCV Library", Dr. Dobb's Journal of Software Tools, (2000).

[11] Zhang, Z., "A Flexible New Technique for Camera Calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, Bd. 22, pp. 1330-1334, (2000).

[12] Kelcey J., and Lucieer, A., "Sensor Correction of a 6-Band Multispecctral Imaging Sensor for UAV Remote Sensing", Remote Sens. (2012), 4, 1462-1493; doi:10.3390/rs4051462.

[13]Smith, G.M., Milton, E.J., "The use of the empirical line method to calibrate remotely sensed data to reflectance", International Journal of Remote Sensing, (1999), vol. 20, no. 13, 2653-2662.

[14] https://www.agisoft.com/pdf/metashape-pro_1_5_en.pdf

[15] Lowe, D, G., "Distinctive Image Features from Scale-Invariant Keypoints", Computer Science Department, University of British Columbia (2004).

[16] Förstner, W., "A Feature based Correspondence Algorithm for Image Matching", Int. Arch. of Photogrammetry, Vol. 26-111, Rovaniemi, Finland, (1986).

[17] Harris, C., Stephens, M., "A Combined Corner and Edge Detector", (1988), Alvey Vision Conference.15.

[18] Brown, D, "Decentering distortion of lenses", Photogrammetric Engineering 32 (3):444-462, (1966).

[19] Fitzgibbon, A.W, "Simultaneous linear estimation of multiple view geometry and lens distortion", In: CVPR, pp.125-132 (2001).

[20] Furukawa, Y., Ponce, J., "Accurate, dense, and robust multi-view stereopsis", VOL. 1,NO. 1, AUGUST (2008).

[21] Shepard, D., "A two dimensional interpolation function for irregularly spaced data", Proceedings of the 1968 *ACM* National Conference. pp. 517–524. *doi*:*10.1145/800186.810616*, (1968).

8 Appendix – Equivalence of OpenCV and Metashape calibration modes

The processing pipeline presented in this document allows for a separate calibration of the intrinsic camera parameters. Both software packages provide slightly differing models describing the lens distortion. Thus, a translation between the parameters is provided. The different terms used by *Metashape* and *OpenCV* are clarified as the following table.

Table 3 Intrinsic matric between *OpenCV* and *Metashape*

Pre-processing (OpenCV)[10]	Metashape[14]
The intrinsic matrix expressed by OpenCV is: $\begin{bmatrix} f_x & 0 & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}$ The parameters are focal length f_x and f_y, principal point O_x and O_y. With distortion taken into consideration, the transition between a point in camera perspective and in 2D image coordinates for OpenCV is: $x=X/Z,\ y=Y/Z,$ $r=\sqrt{x^2+y^2}$ $x'=x\frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^2+k_6r^6}+p_1(r^2+2x^2)+2p_2xy$ $y'=y\frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^2+k_6r^6}+p_2(r^2+2y^2)+2p_1xy$ $u=f_x*x'+O_x$ $v=f_y*y'+O_y$ where: (X,Y,Z) - point coordinates in the local camera coordinate system, (u,v) – projected point coordinates in the image coordinate system (in pixels) Radial distortion coefficients: k_1, k_2, k_3, k_4, k_5, k_6 Tangential distortion coefficients: p_1, p_2	Designated as sub-object "Initial", the intrinsic matrix expressed by Metashape is: $\begin{bmatrix} f+B_1 & B_2 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{bmatrix}$ The parameters are focal length f, affinity corrector B_1, skew B_2, principal point O_x and O_y. Metashape explains principal point $O_{x,y}$ by offset $C_{x,y}$ such as $O_x=W/2+C_x$, $O_y=H/2+C_y$. With distortion taken into consideration, the transition between a point in camera perspective and in 2D image coordinates for Metashape is: $x=X/Z,\ y=Y/Z,$ $r=\sqrt{x^2+y^2}$ $x'=x(1+K_1r^2+K_2r^4+K_3r^6+K_4r^8)+[P_1(r^2+2x^2)+2P_2xy](1+P_3r^2+P_4r^4)$ $y'=y(1+K_1r^2+K_2r^4+K_3r^6+K_4r^8)+[P_2(r^2+2y^2)+2P_1xy](1+P_3r^2+P_4r^4)$ $u=W*0.5+C_x+x'(f+B_1)+y'B_2$ $v=H*0.5+C_y+y'f$ where: (X,Y,Z) - point coordinates in the local camera coordinate system, (u,v) – projected point coordinates in the image coordinate system (in pixels), Radial distortion coefficients: K_1, K_2, K_3, K_4, Tangential distortion coefficients: P_1, P_2, P_3, P_4
When $f_y=f$, $f_x=f+B_1$, $B_2=0$, $k_1=K_1$, $k_2=K_2$, $k_3=K_3$, $p_1=P_2$, $p_2=P_1$ and the other radial and tangential distortion coefficients are 0, resulting (u,v) in both systems are identical.	