Evaluation of low level feature-base methods for creating panoramic mosaics of crop images

Pedro Henrique Soares de Almeida, Lilian Tais de Gouveia, Luciano José Senger

Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, Carlos Cavalcanti Av., 4748, 84030-900, phone +55 42 3220 3097, Brazil. E-mails: pedro.almeida1191@gmail.com, ltgouveia@uepg.br, ljsenger@uepg.br

Abstract: Unmanned aerial vehicles are making farming more efficient, by allowing farmers to manage crops using georeferenced images. Images acquired from aerial vehicles cameras area used to build image mosaics of the crops. Image mosaicking is the alignment of multiple images into larger compositions which represent portions of a 3D scene. Several image mosaicking algorithms have been proposed over the last two decades. Among all, low-level feature detecting algorithms may be invariant to scale and rotation, among other transformations that commonly occur in agricultural images obtained by unmanned aerial vehicles. This study aimed to evaluate low level feature-based mosaicking methods using agricultural images obtained by unmanned aerial vehicles. The Harris corner detector, the FAST corner detector, the SIFT feature detector and the SURF detector were evaluated according to the computational performance and the quality of the generated mosaics. To assess computing performance, were considered factors such as the detected features average per image, the number of images used to compose the mosaic and the processing times. To assess quality, the mosaics generated by each method were used to estimate the Asian soybean rust severity and a comparison with the commercial software Pix4Dmapper was performed. Regarding quality, there was no significant difference and all methods proved to be on the same level. SURF detector used, on average, only 33.1% of the input images to compose the mosaics. The Harris corner detector achieved the best computing performance results. However, in its final mosaic, the usage of the input images was below than 50%. The FAST corner detector presented the best utilization of the input images, but significant discontinuities of objects where observed in the overlap regions of the resulting mosaics. Besides, the FAST presented the worst computing performance. The SIFT feature detector achieved the second-best processing time, the second-best utilization of the input images and built mosaics without discontinuities in overlapped regions.

1 Introduction

Unmanned aerial vehicles UAVs are making farming more efficient, by allowing the farmer to manage crop soil, to fertilize and to control crops diseases more effectively. When equipped with digital cameras, UAVs can collect georeferenced images of crop fields. These images can be used by farmers to make better decisions about farming practice. For instance, images acquired from UVA’s cameras are frequently used to build image mosaics which are used to support the farmer’s decisions [1].

Image mosaicking could be considered as a special case of scene reconstruction where the images are related only by planar homography [2][3]. An application scenario of UAVS in agriculture is to help farmers with crop diseases control. The quantification of damages is a key point in the definition of any disease control strategy [4]. Also, several companies also show interest and focus their efforts on this area through their commercial software packages. UAVs are less affected by cloud cover because they can fly at low altitudes but have problems when capturing high-quality images due to the instability caused by their light weight. Since a single image can cover only a limited area, many images need to be captured, making the image mosaicking process much more difficult. The research community demonstrates real interest in image mosaicking area for both its scientific significance and potential derivatives in real world applications [5]. For instance, Figure 1 shows a mosaic image generated from 50 images collected from an UVA's camera.

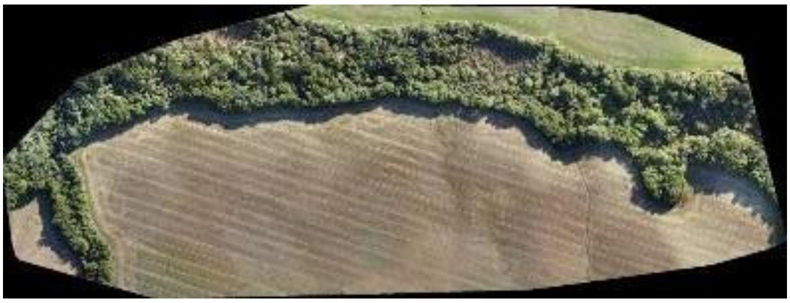


Figure 1: Panoramic mosaic built using 50 from a soy crop located at Ponta Grossa city, Paraná, Brazil.

The objective of this paper is to evaluate 4 classes of feature-based mosaicking methods using crop images obtained from an UAV camera as input. The evaluation was conducted comparing the computing performance and the quality of mosaics created by each method. Three metrics were adopted to assess performance: the average number of features and inliers detected in each image, the number of images used to create the mosaic and the processing time of each method. In order to conduct a qualitative evaluation, the mosaics built were used to estimate the Asian soybean (Glycine max) rust severity and the results were compared with the commercial software Pix4Dmapper[1].

2 Background

Image mosaicking involves the following image processing steps: registration, reprojection, stitching, and blending [5]. These steps are illustrated in Figure 2. The registration step establishes a geometric match between a pair of images describing the same scene. To register a set of images, it is necessary to estimate the geometric transformations that align the images according to a reference image within that set. The set can consist of two or more images taken from a single scene at different times, from different viewpoints, and/or by different sensors. The reprojection refers to the alignment of images to a common coordinate system using geometric transformations. The goal of stitching step is to overlay the aligned images in a larger composition, combining pixel values of overlapping parts and retaining pixels where no overlap occurs. Errors propagated by these steps, due to geometric and photometric misalignments, often result in object discontinuities and seam visibility at the images boundaries. The final merging step tries to minimize this effect and homogenize the overall appearance of the mosaic.

[1] is a photogrammetry software for aerial mapping applications [6].

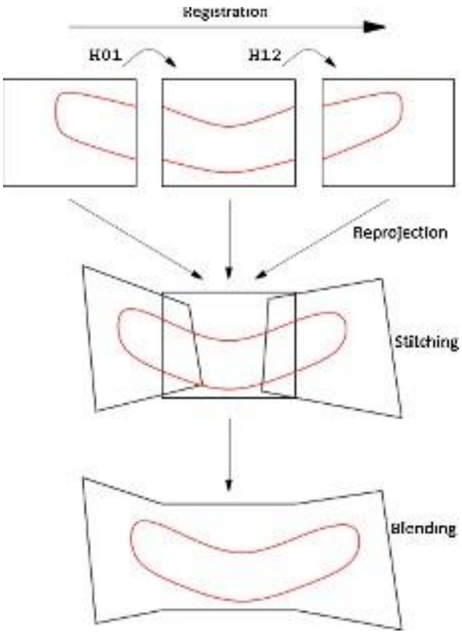


Figure 2: Computing steps when generating a mosaic (source: [5]).

The first step, image registration, is considered the core of image mosaicking and it is done by feature extraction and matching algorithms. Among all, low level feature extraction algorithms are show some advantages over other methods, as speed, robustness, and the availability of creating panoramic image of a non-planar scene with unrestricted camera motion [7, 8].

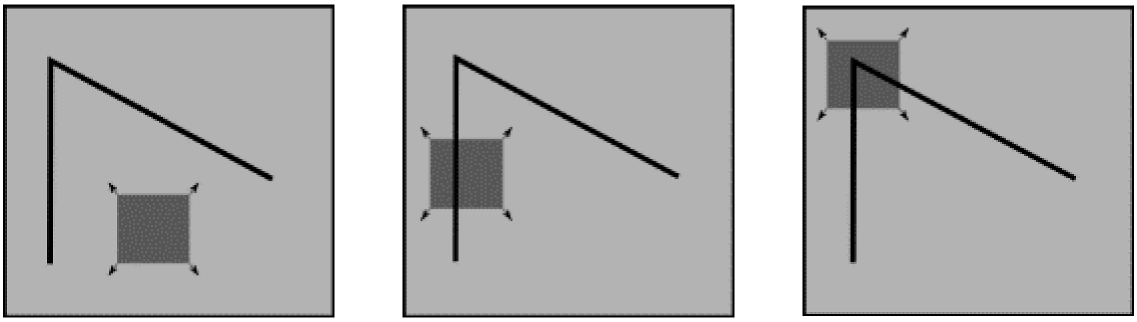
Instead of using all the available data, feature based algorithms try to establish feature points correspondences from all images to be registered. Different features have been adopted, including region, line and point features. In order to obtain an image mosaic, the features are matched by utilizing a correlation measure in the local neighborhood. Low level feature-based algorithms perform image mosaicking even when images large overlapping areas are not available [5]. Low level feature algorithms are also frequently invariant to scaling and rotating, among other transformations. This type of transformations is common in images obtained by cameras.

When camera equipped unmanned aerial vehicles fly over the crop, the images acquired are frequently overlapped and are subject to transformations, due all kind of motion in the camera during images acquisition. According to [5], the main low-level feature extraction algorithms are the following: Harris [9], FAST [10], SIFT [11] and SURF [12].

As the name suggests, the Harris corner detector detects corners in the images as robust low-level features. Initially, a small local detection window is projected into the image (Figure 3). In sequence, the intensity variation resulting from the displacement of this window for a short distance in any direction is determined by the (13):

$$E(U, v) = \sum_i w(x_i, y_i)\ [I(x_i + U, y_i + V) - I(x_i, y_i)]^2 \ (1)$$

where $w(x_i, y_i)$ is the window function for the detection window, (x_i, y_i), $I(x_i, y_i)$ is the image intensity value at the location of Pixel (x_i, y_i), and $I(x_i + U, y_i + V)$ is the intensity with offset (u,v).



(a) (b) (c)

Figure 3: local detection window of intensity variation. (a) No variation in all directions. (b) No variation in edge direction. (c) Meaningful variance in all directions (corner found).

The local texture around the pixel (x_i, y_i) is expressed as an autocorrelation matrix C, as follows:

$$C = \sum_i w(x_i, y_i) \begin{bmatrix} I_{x_i}^2 & I_{x_i} I_{y_i} \\ I_{x_i} I_{y_i} & I_{y_i}^2 \end{bmatrix} \quad (2)$$

where I_{x_i} and I_{y_i} are the first derivative of $I(x_i, y_i)$. Two large autovalues for the matrix C correspond to a corner point. In this case, the center point of the window is characterized as such. For greater robustness, a measure of *cornerness*[2] R is used to eliminate edge points, as Equation (3):

$$R = Det(C) - \alpha Tr^2(C) \quad (3)$$

where $Tr(C)$ is the dash of C and α is within the range $0.04 \leqslant \alpha \leqslant 0.06$. Corner points are detected as local maximums of R above a predefined T threshold.

After the corner points are detected in both images, the matches are established by NCC or by another sum of squares method of the SSD differences. Subsequently, the geometric movement parameters are calculated and the images are deformed according to a global reference image in order to create the mosaic. Mosaic algorithms using the Harris corner detector are computationally simple and almost accurate [5].

Like the Harris corner detector, the FAST corner detector detects corners in the images. This algorithm is computationally more efficient and faster than most other low-level characteristic extraction methods. Consequently, the mosaic methods based on this algorithm are particularly suited for real-time image processing applications.

Initially, a circle of 16 pixels is considered around each pixel candidate to corner. The candidate is considered a corner if there is a set of n contiguous pixels in the circle that are or all clearer than the pixel's candidate intensity plus a threshold, or all darker than the candidate pixel intensity minus a threshold, as illustrated in Figure 4.

[2] the purpose of this measure is to associate a score proportional to howstrongly a region of the image has a particular structure, such as a corner.

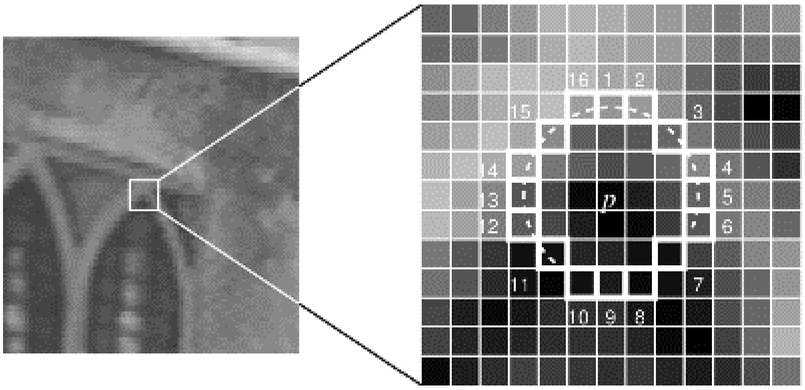


Figure 4: FAST algorithm candidate search. Source: [10]

Choosing an optimal threshold is often a fundamental challenge of the algorithms based on the FAST corner detector. The number *n* is usually set to 12. In order to minimize FAST computing times, a CRF function is used. The CRF function provides the *cornerness* measure of a corner point based on the local neighborhood image intensities [13]. The corners are detected as local maximums for the calculated CRF function over the entire image. After detection, the corner point matching is performed for each pair of images. Then the homograph arrays are calculated and thus the images are projected into a common coordinate system to generate the mosaic.

The SIFT detects distinct characteristics (key points) in images. The algorithm is invariant to the translation, rotation, and scaling transformations in the image domain and robust for moderated perspective transformations and lighting variations. The SIFT algorithm has five main steps: Scale space construction, detection of extremes in scale space, location of key points, assignment of orientation and definition of descriptors to key points [5].

Initially, a scale space is constructed from the repeated convolution of an image using a Gaussian filter, with changes in scale and grouping of outputs in octaves, according to Equation 4 [11]:

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \quad (4)$$

where * is the convolution operator, $G(x,y,\sigma)$ is a Gaussian filter with variable scale σ and $I(x,y)$ is the input image. After the construction of the scale space, the Gaussian difference images are computed from the adjacent Gaussian images in each octave, as follows (Equation 5):

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma) \quad (5)$$

Then, the candidate key points are identified as local extremes of the DoG images on three scales: the current, the immediately above, and the immediately below. In the next step, low-contrast and/or localized key points along edges are discarded using an accurate key-point location. Key points are then assigned to one or more orientations based on the local gradient directions of the image, as Equation $\theta(x,y) = tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$ (6):

$$\theta(x,y) = tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \quad (6)$$

where $\theta(x,y)$ represents the direction of the gradient to $L(x,y,\sigma)$. A set of orientation histograms is formed over the neighbourhoods of each key point. Finally, a normalized vector of 128 dimensions is calculated for each key point as its descriptor [15].

In order to find the corresponding initial key points from two images, the closest neighbor to a key point of the first image is identified from a database of key points of the second image [11,16]. After the initial match, the RANSAC [17] algorithm is used to remove the false matches and calculate the transformation parameters between a pair of images. Finally, the images are deformed using the transformation parameters and combined to generate the mosaic. Image mosaic algorithms based on the SIFT feature detector are considered when combining high-resolution images under a variety of changes (rotation, scaling).

The SURF detector is an invariant feature detector at scale and rotation. Just like the SIFT feature detector, it is also based on scale space theory. However, the SURF detector uses the Hessian array of the entire image to estimate local maxima in different range spaces [18]. The Hessian array [12] of an image *I* with scale σ at any point (*x*,*y*) is defined as follows (Equation 7):

$$H(x,y,\sigma) = \begin{pmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{pmatrix} \tag{7}$$

where $L_{xx}(x,y,\sigma)$, $L_{yy}(x,y,\sigma)$ and $L_{xy}(x,y,\sigma)$ are the convolutions of *I* at point (*x*,*y*) with second-order Gaussian filters $\frac{\partial^2}{\partial_{x^2}}G(x,y,\sigma)$, $\frac{\partial^2}{\partial_{y^2}}G(x,y,\sigma)$ and $\frac{\partial^2}{\partial x \partial y}G(x,y,\sigma)$, respectively.

When calculating the Hessian matrix in each pixel, the operations of Gaussian filters are approximated by operations using box filters, as illustrated in Figure 5. The response in each pixel is calculated as the determinant of the Hessian matrix. Then a threshold and a maximum local detection window of 3×3×3 are used for non-maxima suppression. The local maxima are interpolated into the scale space to get the key points with their location and scale values. In order to assign guidance for each key point, Haar's *wavelet* responses are computed within a circular neighborhood around each key point. A vector isformed by summing all the answers within a window of 60 degrees. The longest vector is assigned as orientation to the key point. In order to assign a vector descriptor to each key point, a square neighborhood region is defined around the key point. This region isdivided into smaller subregions. The sum of the responses of the *wavelet* of Haar from all sub-regions are used to generate a vector descriptor of 64 dimensions [19].

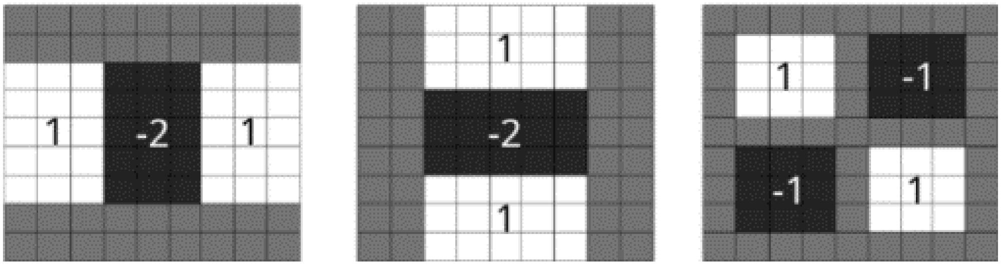


Figure 5: From left to right: approximation of second orderpartial Gaussian derivatives in directions *x*, *y* and *xy*.

After you find the corresponding key points from a pair of images, the RANSAC algorithm is used to eliminate the false matches, as well as to calculate the homography arrays. Once the homography matrices are obtained, the images are deformed and combined to obtain the final mosaic. The continuous advent of new mosaicking techniques in recent years makes it really difficult to choose an appropriate mosaicking algorithm for a specific purpose. Also, crop images introduces some difficult to create mosaics, because of the relative homogeneity of crops in the field. Due to this, it is very important to evaluate the quality of mosaicking algorithms and its computing performance when used to solve precision agriculture issues.

3 Material and Methods

The crop images were acquired from the UAV eBee camera, manufactured by senseFly (Figure 6). Before each UAV flight, 5 GCP measuring 60x60cm were positioned. The geographical coordinates of the central point were obtained using the GNSS receiver Trimble Juno SC. The flights were conducted at an altitude of 530 meters, between 11am and 2pm. They were performed weekly, on the day of the week that presented favorable climate conditions.

The UAV used is equipped with a Sony Cyber-shot DSCWX220 camera with 18.2 megapixels. The resolution of crop images is 15cm/pixel. From the flights, 3 groups of images were obtained. The photographed soybean crop is located at a farm named Fazenda Escola, in the city of Ponta Grossa – PR – Brazil.

Figure 6: UAV used for capturing crop images.

Four farm parcels (A3, A4, B3 and B4) were photographed. Each parcel was subdivided into 4 blocks with 11 crop treatments per block. Parcels A3 and B4 contained the 5969 Nidera cultivar with a population of 14 or 15 plants per meter. Parcels A4 and B3 contained the TMG 7262 cultivar with a population of 10 plants per meter.

The computer program was implemented in C++ programming language. Features available in the OpenCV[3] library (version 3.3.1) were used, following the main ideas from [20, 21]. The computing model steps is detailed in Figure 7. The whole process was organized into 3 main steps: Registration, Calibration and Compositing.

[3] https://opencv.org

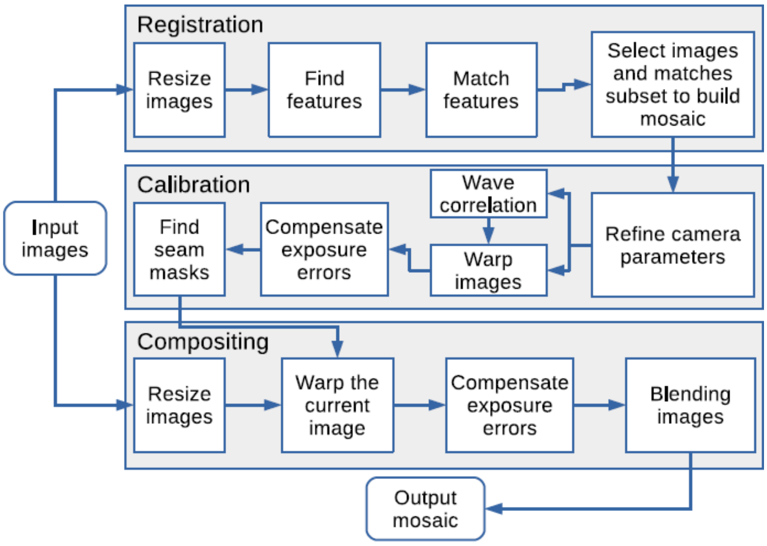


Figure 7: Computer program main steps.

The OpenCV library defines the stitching module and the detail submodule, which contains a set of functions and classes that implement a stitcher[4]. However, the resources available for conduction the registration step in the stitching are not compatible with the purpose of this study, because the Harris corner detector, the FAST corner detector and the SIFT feature detector were not available within this module. Thus, the Registration step was completely reimplemented and the remaining mosaicking steps could be following by using the overall stitching module functions.

The Registration step initially resizes the input images to a medium size, detects and describes the features in each one of them. The Harris corner detector and FAST corner detector do not use descriptors, but detectors. The BRISK [22] descriptor was used. The default openCV detector and descriptor parameters were used. For every feature, the 2 best candidate matches (nearest neighbors) are kept and the ratio test is applied in both directions (for instance, from imageA to imageB and from imageB to imageA).

A ratio test rejects all matches in which the distance ratio between the nearest neighbors is greater than 0.8, which eliminates 90% of the false matches (outliers) while discarding less than 5% of the correct matches (inliers) [11]. Also, in the Registration step, a symmetry test is performed, keeping only features of the imageA and imageB that match to each other simultaneously. Then, the homography between the pair of images is computed using the RANSAC [17] algorithm.

Right after, a refinement is performed and the homography is computed again, but this time only with the obtained inliers. In the sequence, the probabilistic model for image matching verification proposed by [20] is computed. With this model, one can ensure if the matching of the pair of images is valid or not. The maximum number of pairs per image satisfying the probabilistic model was limited to 6, as suggested by [20].

The Calibration step focuses on minimizing differences between an ideal model and the camera-lens system: different camera positions and optical defects such as distortions, exposure, chromatic aberrations, and so on [21]. The parameters are refined globally using bundle adjustment [23], which is a photometric technique to combine multiple images of the same scene into an accurate 3D reconstruction. The aim is to find a globally consistent set of alignment parameters that minimize the mis-registration among all pairs of images [7]. In the sequence, optionally, wave correlation can be

[4] It can be understood as a program that creates mosaics from two or more images

calculated to improve the camera setting. It was not necessary to calculate it for the mosaics created from the groups of images used in this study. Also, in the Calibration step, since mosaicking could be regarded as a special case of scene reconstruction where the images are related only by planar homography [2], a planar map projection is used, and each image is warped. Then, the exposure errors are compensated on each warped image.

Finally, the best areas of attachment for each image (seam masks) are found. The Compositing step uses the results of the previous step, Calibration, combined with the remapping of the images to an output projection. Colors are also adjusted between images to compensate for exposure differences. Images are blended together, and seam-line adjustment is done to minimize the visibility of seams between images [21]. First, each input image is read and, if necessary, resized. Then, these images are warped and compensated for exposure errors. Finally, these images are multiband blended.

The scale of [24] (Figure 8) was used to quantify the rust severity on field. 7 randomly chosen plants in the center of each parcel were analyzed in 3 parts: top, middle and bottom. The final value of the severity was calculated from the average values of the 3 parts and the 7 plants.

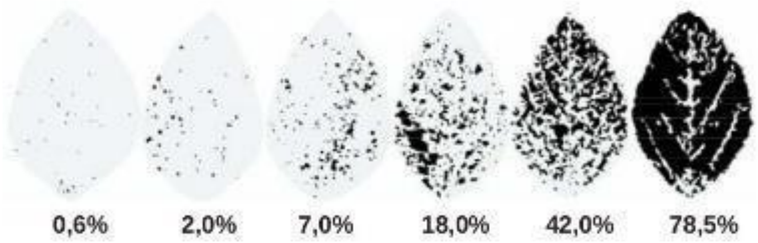


Figure 8: Diagrammatic scale of soybean (Glycine max) rust severity (percentage of diseased leaf area).

An Intel Core i7-2600 CPU 3.40GHz, 24GB RAM and executing Ubuntu 16.04 operating system was used in the experiments. To conduct the computing performance, execution times were recorded. In this context, the execution time describes the interval between the moment when the program is started and the moment when the final mosaic is written to file system. Partial computing times were also recorded. Regarding the Registration step, for better comparing processing times among the methods, the evaluation was organized in two substeps. The first substep (Detect/Describe) describe the execution time it took to detect and describe the features of all images. The second substep (Match) describe the computing time it took to match and filter the features and to form the pairs of images.

The performance evaluation was conducted considering the number of images used to compose the mosaic, the average of features detected in each image, the average of inliers detected in each image, the partial processing times in seconds (s) (Registration, Calibration and Compositing steps) and the total processing time for each method.

3.1. Data extraction from mosaics

With the mosaics created by the implemented program, the software Quantum GIS (version 2.8.13) was used to correct the georeferencing of each one based on the coordinates of the GCP. The software was also used to extract data from mosaics, as shown in Figure 9. The white circles represent the five GCP. The yellow lines represent the divisions of the blocks within each parcel. The light green rectangles delimit the region for data extraction in each treatment. In each region 200 random points were defined, representing approximately 10% of its total area. These points were used to extract the values for RGB bands. The same process was repeated for the mosaics created by the software Pix4Dmapper (version 3.1.23).

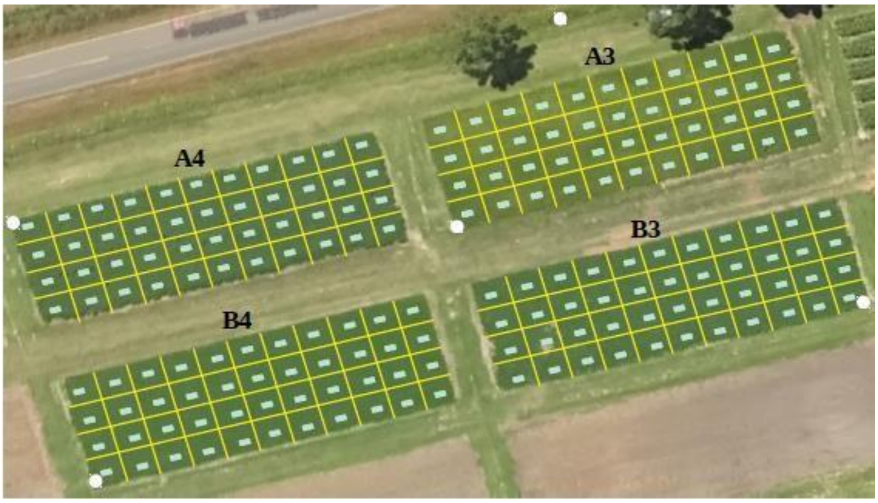


Figure 9: Mosaic and labels used for georeferenced information extracting.

3.2. Data mining

The extracted data were organized in two data bases. The first base contains values referring to parcels A3 and A4 and the second base contains values referring to parcels B3 and B4. The data bases were submitted to statistical tests in software R5 (version 3.3.3). In each base, in order to estimate the rust severity indices, a regression analysis was performed using the SVR algorithm [25] present in e1071 package. The cross-validation method was used to validate the regression models.

The test set was organized in 10 folds and the prediction was applied in each one separately. Finally, to conduct the performance evaluation of the mosaicking methods and the implemented program varying the number of input images, a new group of crop images was used. This group (B) was composed of 50 images that were obtained in an area with a more homogeneous appearance. Although the farm's region was different, the obtaining conditions were the same as those previously described (the same UAV, flight altitude, time, camera, resolution and average overlapping area).

4. Results and Discussion

Figure 10, Figure 11 and Figure 12 show the mosaics created by each method from groups of crop images. Tables Table 1, Table 2 and Table3 present the performance data obtained by each method from these mosaics. It was observed that: a) the average number of detected features directly affected the processing time of the Registration step; b) the number of images used to compose the mosaic directly affected the processing time of the Compositing step and; c) the average number of detected inliers and the number of images used to compose the mosaic affects the Calibration step processing times. The SURF detector was the only one that did not use all available images to compose the mosaic (Tables Table 1 and Table 2). It is due to the probabilistic model: the condition imposed by the probabilistic model for image match verification phase was not satisfied for some pairs of images. Due to this behavior, the processing times of the Calibration and Compositing steps were the lowest among all methods and, consequently, also the lowest total times. However, this apparent performance improvement was not observed in the Registration step. This was due to the elevated average number of features and inliers detected by SURF detector, only smaller than the FAST corner detector.

For the other methods (Tables Table 1, Table 2 and Table 3), one can observe that FAST corner detector had a considerably higher processing time for the Calibration step and Match substep. In the Calibration step, this happened due to the number of detected inliers. For the Match substep, this is explained by the average of detected features, which was near to 13.7 times higher when compared with the lowest average number of features in Tables Table 1, Table 2 and Table 3. However, the processing time for the Detect/describe substep was only 1.6 times above the faster average time from Tables Table 1, Table 2 and Table 3.

FAST corner detector was able to detect features faster than the others, which corroborates with the results obtained by [7], who compared the processing times among several image feature extraction techniques with focus on real time image mosaicking. FAST corner detector was the fastest, immediately followed by Harris corner detector.

SURF detector used all images to compose the mosaic, resulting in the vanishing of the advantage in the processing times presented in Tables Table 1 and Table 2 (Table Table 3). The total execution time was even higher than SIFT feature detector. However, this happened due to the number of detected features and inliers. In [26], the authors compared different image matching techniques against different kinds of transformations and deformations, with a similar number of features detected per image. SURF detector was almost 3 times as fast compared to SIFT feature detector.

Table 1: Performance data from the group of images obtained on 02/15/2017.

		SIFT	SURF	FAST	Harris
Average features		3,083	5,471	25,262	1,651
Average inliers		265	207	666	73
2*Registration	Detect/describe	2.74s	2.34s	3.39s	2.10s
	Match	3.29s	5.69s	67.05s	0.36s
Calibration		4.08s	0.44s	16.01s	3.79s
Compositing		8.67s	4.11s	7.19s	9.26s
Total time		21.05s	14.08s	95.38s	17.90s

Table 2: Performance data from the group of images obtained on 02/21/2017.

		SIFT	SURF	FAST	Harris
Number of images		6	2	6	6
Average features		3,236	6,204	28,780	3,087
Average inliers		393	364	1,043	158
2*Registration	Detect/describe	3.68s	3.32s	4.59s	2.87s
	Match	5.56s	11.56s	139.14s	1.68s
Calibration		7.99s	0.13s	25.12s	12.88s
Compositing		8.87s	2.44s	8.84s	8.63s
Total time		27.93s	18.65s	179.45s	27.76s

Table 3: Performance data from the group of images obtained on 03/13/2017.

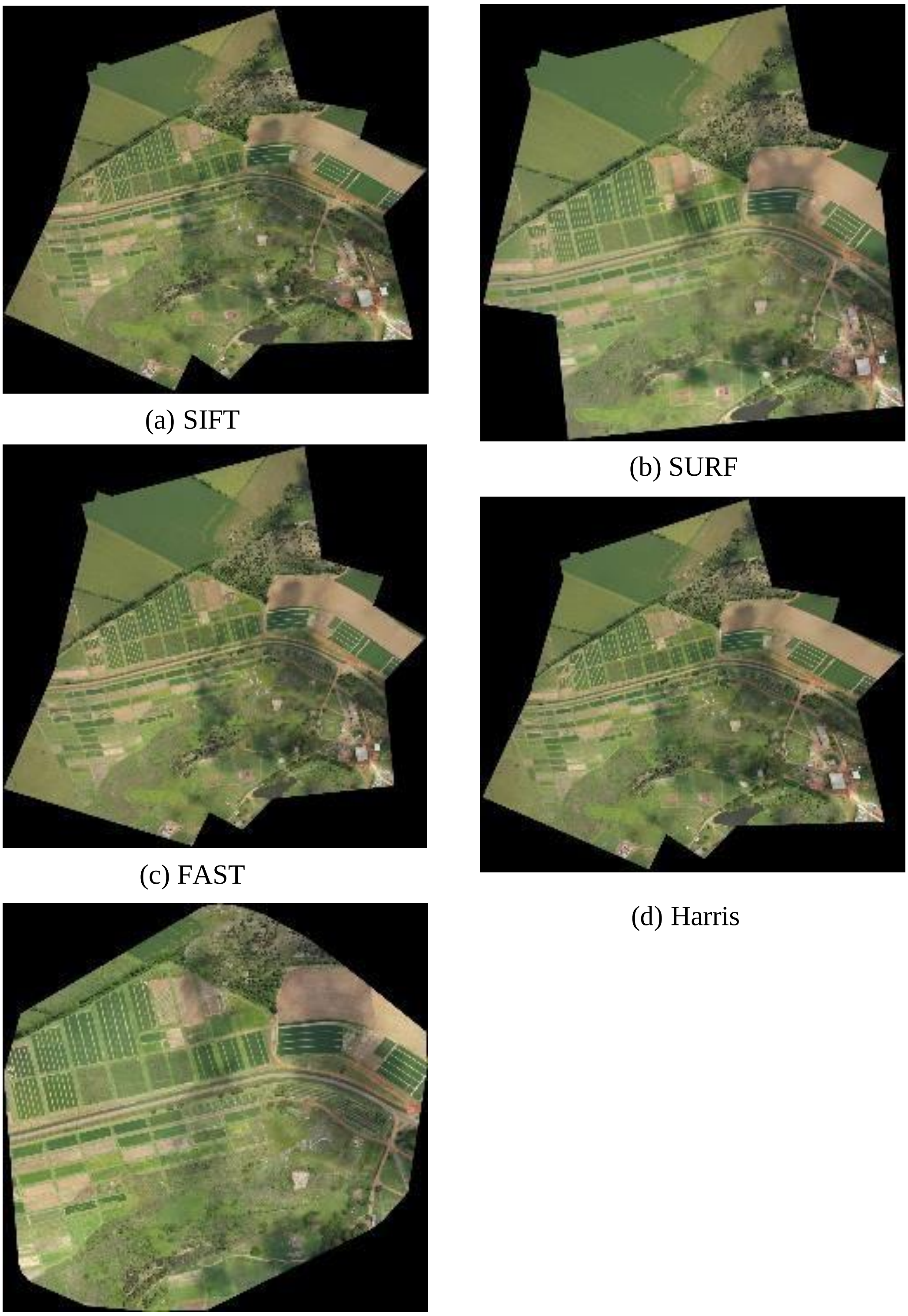
		SIFT	SURF	FAST	Harris
Number of images		6	6	6	5
Average features		2,570	4,987	23,363	915
Average inliers		357	598	1,487	56
2*Registration	Detect/describe	3.23s	2.74s	3.79s	2.35s
	Match	3.57s	8.11s	94.65s	0.22s
Calibration		19.86s	18.59s	52.85s	5.20s
Compositing		7.32s	7.34s	7.43s	8.43s
Total time		35.69s	38.52s	160.50s	18.56s

4.1. Rust severity results

Tables Table 4 and Table 5 show the rust severity data obtained by each method from the mosaics created with the crop of images. The Correlation Coefficient (R) and the RMSE are presented, according to the date information from crop images. One can observe from the data of 02/15/2017 (Table 5) that the calculated R for SURF detector was reasonably lower than the others. This happened, specifically in this case, due to the low level of detail (low resolution) of the generated mosaic, especially in the region of the parcels B3 and B4. The level of detail is associated with the spatial resolution, which can directly affect the results obtained from the data extracted from the image.

In [27], the authors conducted a study comparing the impact of spatial resolution of images on the detection of HLB disease, which attacks citrus plants. Resolutions of 50cm/pixel and 5.45cm/pixel were compared. The authors found that the data set based on the higher resolution images produced a better accuracy in the classification (67-85%) and less false negatives (7-32%) than the corresponding data set based on the images with lower resolution (61-74% and 28-45%, respectively).

Despite the low R calculated for SURF detector in Table 5, a single-factor ANOVA with significance level of 95% was performed with both R values and RMSE values. The ANOVA test shows there is no significant difference in the means. This indicates that the final quality of the mosaics generated by the evaluated methods and by the software *Pix4Dmapper* was on the same level.



(a) SIFT

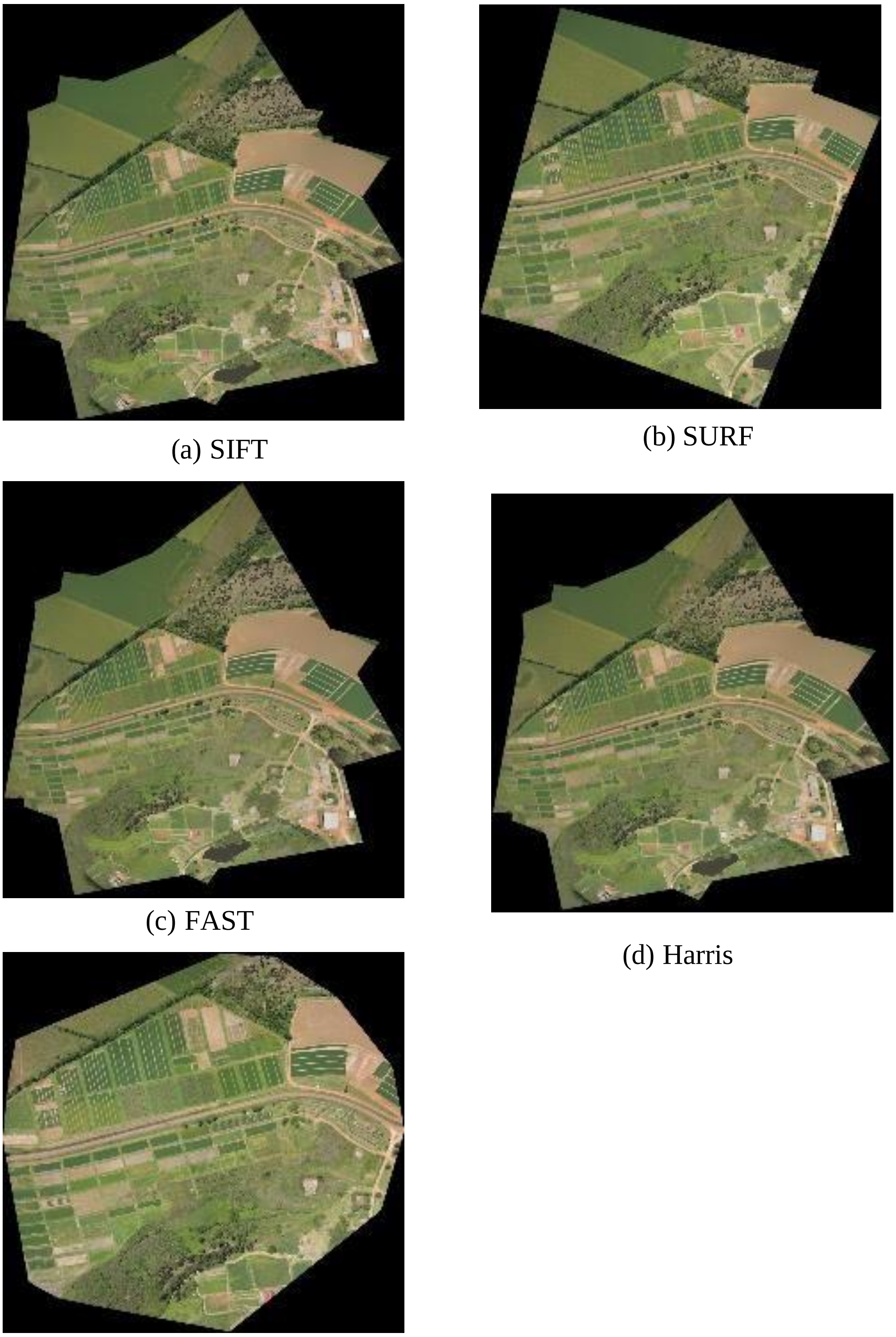
(b) SURF

(c) FAST

(d) Harris

(e) Pix4Dmapper

Figure 10: Mosaics created from the group of images obtained on 02/15/2017.



(a) SIFT

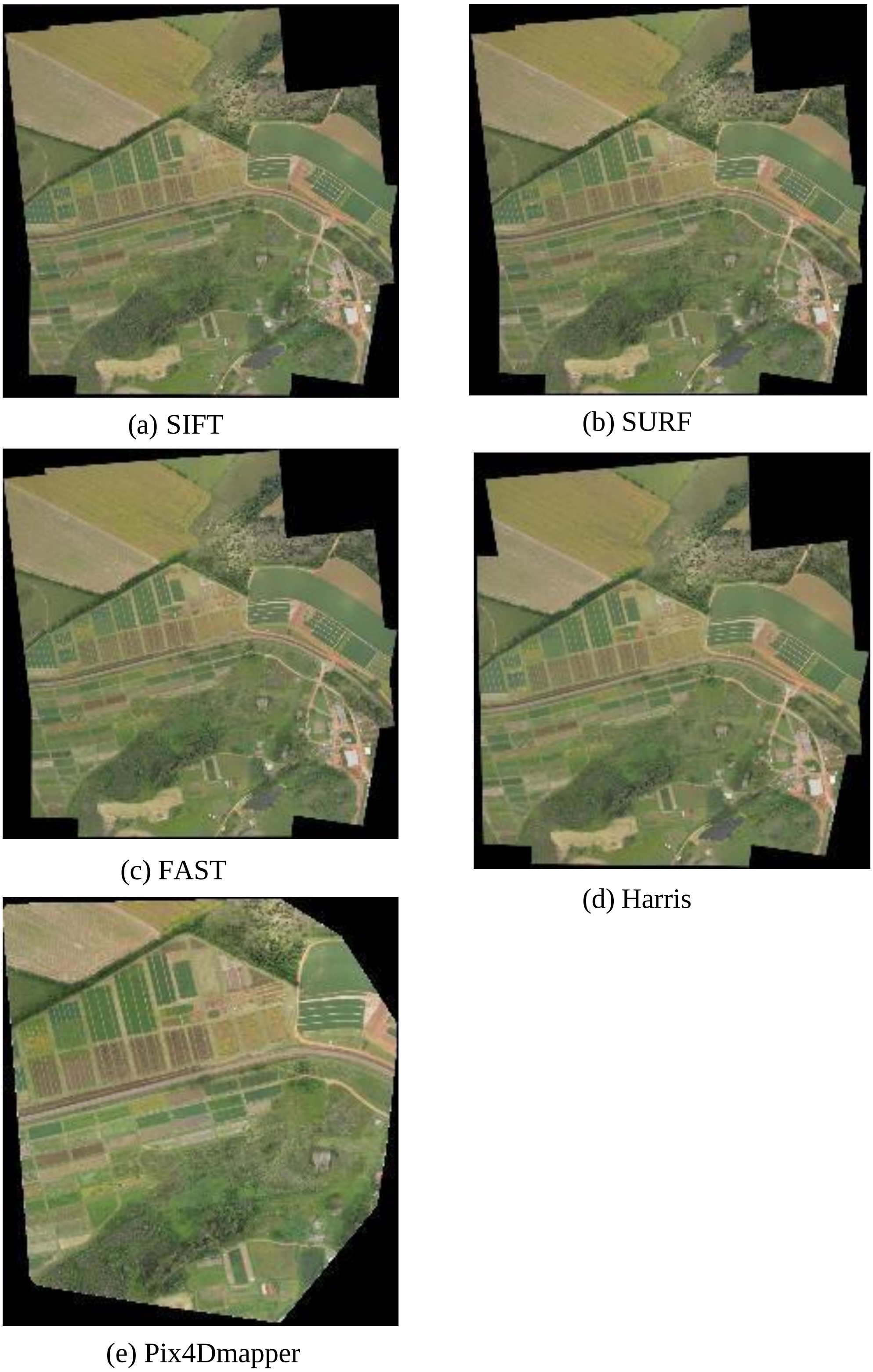
(b) SURF

(c) FAST

(d) Harris

(e) Pix4Dmapper

Figure 11: Mosaics created from the group of images obtained on 02/21/2017.



(a) SIFT

(b) SURF

(c) FAST

(d) Harris

(e) Pix4Dmapper

Figure 12: Mosaics created from the group of images obtained on 03/13/2017.

Table 4: Asian soybean rust severity data from parcels A3 and A4.

	02/15/2017		02/21/2017	
	R	**RMSE**	**R**	**RMSE**
SIFT	0.83	6.10	0.84	5.89
SURF	0.86	5.56	0.86	5.54
FAST	0.83	6.09	0.85	5.76
Harris	0.83	5.98	0.85	5.67
Pix4Dmapper	0.85	5.71	0.88	5.17

Table 5: Asian soybean rust severity data from parcels B3 and B4.

	02/15/2017		02/21/2017		03/13/2017	
	R	**RMSE**	**R**	**RMSE**	**R**	**RMSE**
SIFT	0.80	4.03	0.78	4.94	0.75	11.68
SURF	0.69	4.59	0.80	4.83	0.76	11.46
FAST	0.80	3.90	0.79	4.86	0.80	10.65
Harris	0.81	3.91	0.77	5.11	0.77	11.35
Pix4Dmapper	0.82	3.74	0.81	4.69	0.79	10.80

4.2. Performance results

Figure 13 shows the mosaics created by each method using the 50 images per mosaic and the Figure 1 show the mosaic created using the Pix4Dmapper software. Table 6 present the performance data obtained by each method.

It is possible to observe that none of the methods used the total number of images to compose the mosaic. FAST corner detector obtained the best use with 48 images. In spite of this, the alignment of the images in the final mosaic was not ideal, causing discontinuities of objects. SIFT feature detector obtained the second best use with 41 images. Harris corner detector used 26 images and SURF detector achieved the worst use, with only 3 images.

Besides the fact that the area used had a more homogeneous appearance, which could make it difficult to match features and, consequently, to form pairs of images. This can be explained due to the poor performance that SURF detector has under certain transformations (e.g. color, illumination), which is common in crop images obtained by UAV; and Harris corner detector is good only for moderate changes in scale and rotation, situations that are easily overcome, especially in images obtained by UAV.

On the other hand, at least related to crop images obtained by UAV, these results, coupled with the poor results obtained by SURF detector in Tables Table 1 and Table 2, go against what is stated by [5]. The average overlapping area was around 90% and the methods were not always able to use all crop images to compose mosaics.

Table 6: Performance data from the group composed of 50 images.

		SIFT	SURF	FAST	Harris
Number of images		41	3	48	26
Average features		4,55	6,19	32,021	6,401
Average inliers		56	53	177	37
2*Registration	Detect/describe	26.63s	24.30s	34.07s	18.40s
	Match	806.24s	867.51s	10,599s	563.88s
Calibration		1,559s	0.36s	8,494s	353.79s
Compositing		46.96s	3.13s	60.58s	15.64s
Total time		2,44s	896.56s	19,19s	953.09s

5. Conclusion

This study presented a comparative evaluation of 4 low level feature-based image mosaicking techniques. Images acquired from UAVs cameras were used and the evaluation was made according to the computational performance and the quality of image mosaics. In order to evaluate performance, the average number of features and inliers detected per image, the number of images used to compose the mosaic and the processing time were taken into account. To evaluate quality, the mosaics generated by each method were used to estimate the Asian soybean (*Glycine max*) rust severity and a comparison with the commercial software *Pix4Dmapper* was performed.

The comparison made between the estimates of the Asian soybean rust severity calculated by the software *Pix4Dmapper* and the other methods demonstrated that the quality of the generated mosaics was at the same level. The ANOVA methodology was used was applied at a confidence value of 95%. There is not difference among group values (rusty severity values) and the results did not show significant differences.

The SURF detector obtained the worst performance among all methods and was not able to match most pairs of images in 3 of the 4 groups of images tested. In these cases, an average of only 33.1% of the input images was used to compose the final mosaic. Thus, presented itself as an inadequate solution for agricultural images obtained by UAV.

Harris corner detector proved to be the fastest solution among all methods and could be reliable for small groups of agricultural images from regions with a more heterogeneous appearance. It was 7.27% faster to compose the final mosaic in cases where all images were used (Tables Table 1 and Table 2). However, in its final mosaic, the use of the input images was poor, being below 50%.

The FAST corner detector, due to the high average number of detected features and inliers, obtained a better use of the number of images in the composition of the final mosaic. However, its processing time was considerably higher, becoming 6.42 times slower than the slowest solution among other methods in cases where all images were used to compose the mosaic (Table 2). In addition, significant discontinuities of objects occurred in its final mosaic. Thus, presented itself as an inadequate solution for agricultural images of regions with a more homogeneous appearance.

The SIFT feature detector presented itself as the most suitable method. It obtained the second best processing time and the second best use of the input images to compose the mosaics, using 82% of them in the worst case (Table 6). In addition, there were no problems with object discontinuities. The SIFT feature detector is efficient for high resolution images and offers invariance to several transformations. Therefore, it proved to be a reliable solution for agricultural images obtained by UAVs.

Acknowledgements

This paper is based upon work supported by CAPES. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the CAPES. We would like to thank the INFOAGRO laboratory staff for providing the UAV images used in this study and for their assistance during the process of estimating the Asian soybean rust severity. Thanks to UEPG and to High Performance Computing Lab (LCAD/UEPG) for providing the computing infrastructure used in the experiments of this research work.

References

[1] LI, Z.; ISLER, V. (2016). Large scale image mosaic construction for agricultural applications. *IEEE Robotics and Automation Letters*, 1(1) , pages 295–302.

[2] GHOSH, D. et al. (2012) Quantitative evaluation of image mosaicing in multiple scene categories. *IEEE International Conference on Electro/Information Technology*, pages 1-6.

[3] ELIBOL, A. et al. (2017). Fast underwater image mosaicing through submapping. *Journal of Intelligent and Robotic Systems*, 85(1), pages 167–187.

[4] HIKISHIMA, M. et al. (2010). Quantificação de danos e relações entre severidade, medidas de refletância e produtividade no patossistema ferrugem asiática da soja. *Tropical Plant Pathology*, 35, pages 96–103.

[5] GHOSH, D.; KAABOUCH, N. (2016). A survey on image mosaicing techniques. *Journal of Visual Communication and Image Representation*, v. 34, p. 1–11, 2016. ISSN 1047-3203.

[6] KAKAES, K. et al. (2015). Drones and aerial observation: New technologies for property rights, human rights, and global development. *New America*, 2015.

[7] ADEL, E.; ELMOGY, M.; ELBAKRY, H. (2014). Real time image mosaicing system based on feature extraction techniques. *9th International Conference on Computer Engineering Systems*, pages 339–345.

[8] LI, A. et al. (2017). An improved FAST+SURF fast matching algorithm. Procedia Computer Science, v. 107, p. 306 – 312. Advances in Information and Communication Technology: Proceedings of 7th International Congress of Information and Communication Technology.

[9] HARRIS, C.; STEPHENS, M. A. (1988). Combined corner and edge detector. In: Fourth Alvey Vision Conference, pages 147-151, Manchester, UK.

[10] ROSTEN, E., DRUMMOND, T. (2006) Machine Learning for High-Speed Corner Detection. In: Leonardis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg

[11] LOWE, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), pages 91–110.

[12]BAY, H.; TUYTELAARS, T.; GOOL, L. V. (2006). SURF: Speeded up robust features. *9th European Conference on Computer Vision*, pages 404–417.

[13]JOSHI, H.; SINHA, M. K. (2013). A survey on image mosaicing techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(2), pages 365–9.

[14]GAO, G.; JIA, K. A (2007). New image mosaics algorithm based on feature points matching. *Second International Conference on Innovative Computing, Information and Control*, pages 471.

[15]GHOSH, D.; KAABOUCH, N.; FEVIG, R. A. (2014). Robust spatial-domain based super-resolution mosaicing of cubesat video frames: Algorithm and evaluation. *Computer and Information Science*, 7(2).

[16]LIQIAN, D.; YUEHUI, J. (2010). Moon landform images fusion and mosaic based on sift method. *International Conference on Computer and Information Application*, pages 29–32.

[17]FISCHLER, M. A.; BOLLES, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *ACM communications*, 24(6), pages 381–395.

[18] YANG, L. et al. (2011). A research of feature-based image mosaic algorithm. *International Congress on Image and Signal Processing*, pages 846–849.

[19]BIND, V. S. (2013). Robust Techniques for Feature-based Image Mosaicing. Phd thesis, National Institute of Technology Rourkela.

[20]BROWN, M.; LOWE, D. G. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1), pages 59–73.

[21]GARCIA, G. B. et al. (2015). *Learning Image Processing with OpenCV*. Packt Publishing Ltd, 2015.

[22]LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. *International Conference on Computer Vision*, pages 2548–55.

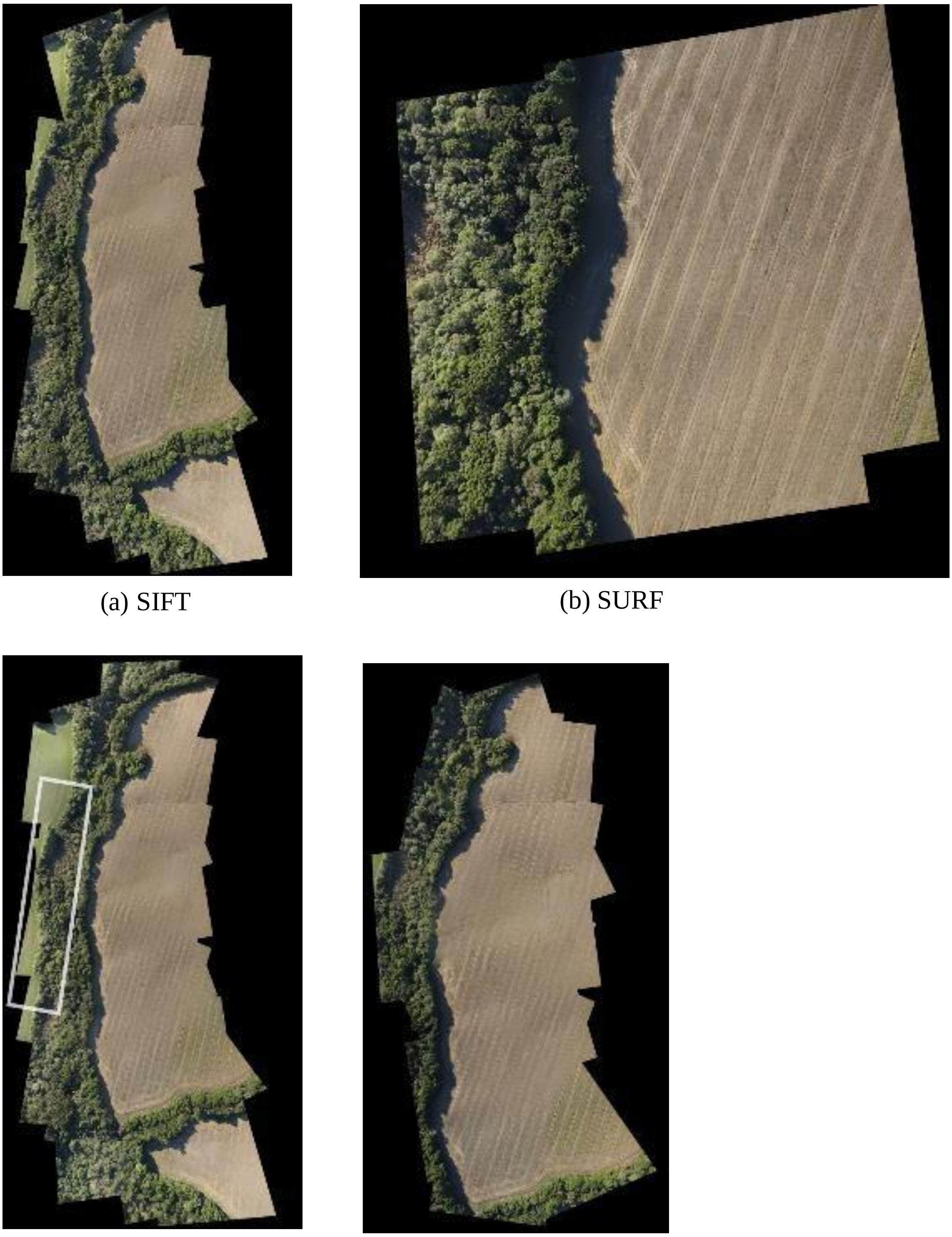
[23]TRIGGS, B. et al. (1999). Bundle adjustment—a modern synthesis. *International workshop on vision algorithms*, pages 298–372.

[24]GODOY, C. A. V.; KOGA, L. J.; CANTERI, M. G. (2006). Diagrammatic scale for assessment of soybean rust severity. *Fitopatologia Brasileira*, 31, pages 63–68.

[25]VAPNIK, V.; GOLOWICH, S. E.; SMOLA, A. J. (1997). Support vector method for function approximation, regression estimation and signal processing. *Advances in Neural Information Processing Systems*., pages 281–287.

[26]KARAMI, E.; PRASAD, S.; SHEHATA, M. Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images. Proceedings of Newfoundland Electrical and Computer Engineering Conference, St. John’s, Canada, 2015.

[27]GARCIA-RUIZ, F. et al. Comparison of two aerial imaging platforms for identification of huanglongbinginfected citrus trees. Computers and Electronics in Agriculture, v. 91, p. 106–115, 2013. ISSN 0168-1699.



(a) SIFT

(b) SURF

(c) FAST

(d) Harris

Figure 13: Mosaics created from the group composed of 50 images.