ACTAS

DE LAS

XXXVIII Jornadas de Automática

Gijón · Palacio de Congresos · 6, 7 y 8 de Septiembre de 2017

CEA 50 ANIVERSARIO

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

CEA
Comité Español de Automática

Colabora
Gijón
Convention Bureau

Actas de

XXXVIII
Jornadas de Automática

Servicio de Publicaciones de la Universidad de Oviedo
Campus de Humanidades. Edificio de Servicios. 33011 Oviedo (Asturias)
Tel. 985 10 95 03 Fax 985 10 95 07
http: www.uniovi.es/publicaciones
servipub@uniovi.es

DL AS 2749-2017

ISBN: 978-84-16664-74-0

Prefacio

Las *Jornadas de Automática* se celebran desde hace **40 años** en una universidad nacional facilitando el encuentro entre expertos en esta área en un foro que permite la puesta en común de las nuevas ideas y proyectos en desarrollo. Al mismo tiempo, propician la siempre necesaria colaboración entre investigadores del ámbito de la Ingeniería de Control y Automática, así como de campos afines, a la hora de abordar complejos proyectos de investigación multidisciplinares.

En esta ocasión, las Jornadas estarán organizadas por la Universidad de Oviedo y se han celebrado del 6 al 8 de septiembre de 2017 en el Palacio de Congresos de Gijón, colaborando tanto la Escuela Politécnica de Ingeniería de Gijón (EPI) como el Departamento de Ingeniería Eléctrica, Electrónica de Computadores y de Sistemas del que depende el Área de Ingeniería de Sistemas y Automática.

Además de las habituales actividades científicas y culturales, esta edición es muy especial al celebrarse el **50 aniversario de la creación de CEA**, Comité Español de Automática. Igualmente este año se conmemora el 60 aniversario de la Federación Internacional del Control Automático de la que depende CEA. Así se ha llevado a cabo la presentación del libro que se ha realizado bajo la coordinación de D. Sebastián Dormido, sobre la historia de la Automática en España en una sesión en la que han participado todos los ex-presidentes de CEA conjuntamente con el actual, D. Joseba Quevedo.

Igualmente hemos contado con la presencia de conferenciantes de prestigio para las sesiones plenarias, comunicaciones y ponencias orales en las reuniones de los 9 grupos temáticos, contribuciones en formato póster. Se ha celebrado también el concurso de CEABOT, así como una nueva Competición de Drones, con el ánimo de involucrar a más estudiantes de últimos cursos de Grado/Máster.

En el marco de las actividades culturales programadas se ha podido efectuar un recorrido en el casco antiguo situado en torno al Cerro de Santa Catalina y visitar la Laboral.

Gijn, septiembre de 2017

Hilario López
Presidente del Comité Organizador

Program Committee

Antonio Agudo	Institut de Robòtica i Informàtica Industrial
Rosa M Aguilar	University of La Laguna.
Luciano Alonso	University of Cantabria
Ignacio Álvarez García	Universidad de Oviedo
Antonio Javier Artuñedo García	Centre for Automation and Robotics (CSIC-UPM)
José M. Azorín	Miguel Hernandez University of Elche
Pedro Balaguer	Universitat Jaume I
Antonio Javier Barragán Piña	Universidad de Huelva
Alfonso Baños	Universidad de Murcia
Guillermo Bejarano	University of Seville
Gerardo Beruvides	Centro de Automática y Robótica
Carlos Bordons	University of Seville
Jose Manuel Bravo	University of Huelva
Jose Luis Calvo-Rolle	University of A Coruña
Fernando Castaño Romero	Centro de Automática y Robótica (UPM -CSIC)
José Luis Casteleiro-Roca	University of Coruña
Alvaro Castro-Gonzalez	Universidad Carlos III de Madrid
Ramon Costa-Castelló	Universitat Politècnica de Catalunya
Abel A. Cuadrado	University of Oviedo
Arturo De La Escalera	Universidad Carlos III de Madrid
Emma Delgado	Universidad de Vigo
Jose-Luis Diez	Universitat Politecnica de Valencia
Manuel Domínguez	Universidad de León
Juan Manuel Escaño	Universidad de Sevilla
Mario Francisco	University of Salamanca
Maria Jesus Fuente	Universidad de Valladolid
Juan Garrido	Universtiy of Cordoba
Antonio Giménez	Universidad de Almeria
Evelio Gonzalez	Universidad de La Laguna
José-Luis Guzmán	Universidad de Almería
Rodolfo Haber	Center for Automation and Robotics (UPM-CSIC)
César Ernesto Hernández	Universidad de Almería
Eloy Irigoyen	UPV/EHU
Agustin Jimenez	Universidad PolitÃcnica de Madrid
Emilio Jiménez	University of La Rioja
Jesus Lozano	Universidad de Extremadura
Jorge Luis Madrid	Centro de Automática y Robótica
Luis Magdalena	Universidad Politécnica de Madrid
David Martin Gomez	Universidad Carlos III de Madrid
Fernando Matia	Universidad Politecnica de Madrid
Joaquim Melendez	Universitat de Girona
Juan Mendez	Universidad de La Laguna
Luis Moreno	Universidad Carlos III de Madrid
María Dolores Moreno Rabel	Universidad de Extremadura
David Muñoz	Universidad de Sevilla
Antonio José Muñoz-Ramirez	Universidad de Málaga
Jose Luis Navarro	Universidad Politecnica de Valencia
Manuel G. Ortega	University of Seville
Andrzej Pawlowski	UNED
Mercedes Perez de La Parte	University of La Rioja
Ignacio Peñarrocha	Universitat Jaume I de Castelló, Spain
José Luis Pitarch	Universidad de Valladolid

Daniel Pérez	University of Oviedo
Emilio Pérez	Universitat Jaume I
Juan Pérez Oria	Universidad de Cantabria
MiguelÁngel Ridao	Universidad de Sevilla
Gregorio Sainz-Palmero	Universidad de Valladolid
Antonio Sala	Universitat Politecnica de Valencia
Ester Sales-Setién	Universitat Jaume I
Jose Sanchez	UNED
Javier Sanchis Saez	Universitat Politecnica de Valencia (UPV)
José Pedro Santos	ITEFI-CSIC
Matilde Santos	Universidad Complutense de Madrid
Alvaro Serna	University of Valladolid
José Enrique Simó	Universidad Politécnica de Valencia
José A. Somolinos	ETS I Navales. Universidad Politecnica de Madrid
Fernando Tadeo	Univ. of Valladolid
Alejandro Tapia	Universidad de Loyola Andalucía
David Tena	Universitat Jaume I
Jesús Torres	Universidad de La Laguna
Pedro M. Vallejo	Universidad de Salamanca
Guilherme Vianna	Universidad de Sevilla
Alejandro Vignoni	AI2 - UPV
Ramón Vilanova	UAB
Francisco Vázquez	Universidad de Cordoba
Jesús M. Zamarreño	University of Valladolid

Revisores Adicionales

Al-Kaff, Abdulla

Balbastre, Patricia
Beltrán de La Cita, Jorge
Bermudez-Cameo, Jesus
Blanco-Claraco, Jose-Luis
Blanes, Francisco
Bonin-Font, Francisco

Cancela, Brais

Ferraz, Luis

Garita, Cesar
Gimenez, Antonio
Gruber, Patrick
Guindel, Carlos

Hernandez Ruiz, Alejandro
Hernandez, Daniel

Jardón Huete, Alberto

López, Amable

Marin, Raul
Marín Plaza, Pablo
Mañanas, Miguel Angel
Morales, Rafael
Moreno, Francisco-Angel

Nuñez, Luis Ramón

Ponz Vila, Aurelio
Posadas-Yague, Juan-Luis
Poza-Luján, Jose-Luis
Pumarola, Albert

Raya, Rafael
Revestido Herrero, Elías
Rocon, Eduardo
Ruiz Sarmiento, José Raúl
Ruiz, Adria

Torres, Jose Luis

Vaquero, Victor

Table of Contents

Ingeniería de Control

Automar

Bioingeniería

Control Inteligente

Educación en Automática

Sistemas de Tiempo Real

Visión por Computador

Real-Time Image Mosaicking for Mapping and Exploration Purposes

Abdulla Al-Kaff, Juan Camilo Soto Triviño, Raúl Sosa San Frutos,
Arturo de la Escalera and José María Armingol Moreno

Intelligent Systems Lab, Universidad Carlos III de Madrid, Madrid, Spain
akaff@ing.uc3m.es, camilosotto@gmail.com, rsosasanfrutos@gmail.com,
escalera@ing.uc3m.es armingol@ing.uc3m.es

Abstract

In the last decade, building mosaic images become an active field in several computer vision and graphic applications. In this paper, a panoramic image construction using monocular camera is proposed. In this approach, SURF algorithm is used to extract the keypoints in order to obtain reliable results for real-time applications. In addition, based on the homography between the panoramic and the new image, the rotation matrix is obtained, and the new image can be projected on a plane parallel to panorama. Finally, image illumination is compensated over the whole image and the calculation of the pixels contributed by each frame in the overlapping areas. The proposed approach has been verified with real flights, and the obtained results show the robustness of constructing panoramic image with minimal loosing in the information, furthermore, the results prove the ability of the proposed approach to create panoramic images in real-time applications.

Keywords: Image mosaic; Panorama; UAV; Exploration; Mapping.

1 INTRODUCTION

The field of Unmanned Aerial Vehicles (UAVs) has been typically limited to and supported by the defense and military industries, this is due to the cost and the complexity of designing, building and operating these vehicles. Recently, with the developments in microelectronics and the increase of computing efficiency, micro unmanned aerial vehicles (MAVs) have encountered a significant focus among the robotics research community. Moreover, because of their ability to operate in remote and dangerous situations, Vertical Take-Off and Landing (VTOL) rotor-craft systems are increasingly used in several civilian and scientific applications; such as surveying and mapping, rescue operation in disasters [1, 2], spatial information acquisition, inspection [3, 4], animal protection [5], agricultural crops monitoring [6], or manipulation and transportation [7]. These capabilities proposed the advantage to substitute the human operators in the risky and hazard environments.

Aerial imagery or aerial filming is considered one of the basic and demanding application; such as filming sports games and events. With the advances in computer vision algorithms and sensors, the concept of using aerial images just for photography and filming was changed to be used widely in more complex applications; such as thematic and topographic terrains mapping [8, 9]; exploration of unreachable areas; such as rivers [10] or forests [11]; surveillance purposes [12].

One of the main tasks in infrastructure inspection and topographical mapping missions is the construction of a panoramic image; in order to cover the whole area under inspection. For this purpose, image mosaicking is required for stitching all the image sequence, and the resulting panoramic image should be as most as possible to the original images without losing any information.

In this paper, a panoramic reconstruction approach for real-time applications, based on monocular images is proposed. This method builds panoramas of infrastructure surfaces and aerial images based on the displacements of the camera. The reconstruction is created by adding the latest captures image to the resulting panorama. This approach is based on the homography estimated from the matched feature points between the panoramic and the new image. Thereafter, the rotation matrix is obtained, and the new image can be projected on a plane parallel to the one of the resulting panorama. The scale and positioning of the latest frame is obtained by comparing the pixels of the feature points. The step prior to the final smooth stitching, is the illumination compensation over the whole image and calculation of the pixels contributed by each frame in the overlapping areas.

The remainder of this paper is organized as follows; section 2 introduces the state-of-the-art work related to image mosaicking. Section 3 presents the proposed image mosaicking algorithm, then section 4 discusses the experimental results. Finally, in section 5 conclusions are summarized.

2 RELATED WORK

There are a number of related works that have similar applications to the proposed approach, nevertheless, they all have slight differences. In this section, these works are depict and analyzed in detail pointing out the strengths and weaknesses of each one of them compared to the proposed approach.

This section is divided into two subsections; in the first one, a review of how regular panoramic images are usually formed. The second one focuses on image mosaic from UAVs to form land views from the sky as well as of large building structures. Additionally, the main differences of these methods to the proposed approach is identified.

2.1 PANORAMIC IMAGES

The panoramas build large and detailed images from several overlapping frames of the same scene [13]. Generally, panoramic images form the mosaic under the assumption of a rotating camera on a fixed position. This means that the matrix transformation of the image is the same in all the captures. Comparing this to the proposed approach, it is noted that it is the key aspect that differs from these methods, However, similar to the other works, the proposed approach assumes all the images are in the same plane.

The works reviewed explain the main steps taken to create a panoramic image [14, 15], or a 360° immerse view [16]. The basic steps carried out to create an image panorama are the following: firstly the overlapping area is detected, from this area, the blending of two frames is created consecutively in a seamless manner.

The main challenges encountered by these articles in the formation of an image mosaic are the following: large amount of data, noise, camera and object motion, vignetting and lens distortion.

2.2 IMAGE MOSAIC IN UAVS

Mosaic from aerial images differs from panoramic images in the aspect mentioned before - all the images are in the same plane. Some image mosaicking techniques presented in [17, 18, 19, 20, 21].

The general steps for automatic image mosaicking are: image pre-process, feature extraction and matching, transformation model construction, transformation coordinate unification and image stitching [17]. The algorithms used to form a panorama from aerial images, usually used for ortho-mapping [18], however, these techniques are used in the formation of panoramic images of buildings and structures for construction and survey purposes.

One algorithm for automatic image mosaicking is proposed in [19], at which, the mosaic from the Unmanned Aerial Vehicle Images (UAVI) are generated; according to the following steps: Frame preprocessing, feature point detection and multi scene stitching.

Image mosaic is also produced based on Structure From Motion (SFM) methods [20]. The steps are similar to the previous work, using SFM for the camera parameters and 3D coordinate calculations.

Similarly, for long-term mosaicking, a radial distortion accumulates error, in [21], a iterative algorithm based on geometrical constraints is proposed; to compensate this error and create panoramas with more than 1500 frames.

Additionally, Eschmann *et al* presented an algorithm; where a multi-copter creates a panorama of a building as well as including a crack detection system based on computer vision [22]. The process is divided into two stages: data acquisition and digital post-processing. In the data acquisition phase, the UAV is controlled manually because this process relies on GPS data, which is insufficient for near buildings flights. Besides, the data for the further image stitching can be collected either automatically, with a given rate, or manually where certain parameters as the zoom may be adjusted. Furthermore, for a proper performance of the algorithm, the UAV must follow a predefined patter of horizontal strips.

3 PROPOSED ALGORITHM

In order to understand the sequence used to create the panoramic image, it is necessary to comprehend the methods, formulas and assumptions behind this project. The algorithm is developed for objects such like large structures or buildings, although it may also be used for ground reconstruction. Ideally, the camera should stay parallel to the planar object being studied, without rotating and keeping the same distance to it. Later, this supposition will not be maintained as there is some noise coming from the UAV flight. This noise will be treated by transforming the images.

The proposed panoramic image reconstruction is designed to be on-board since a live monitoring reconstruction is possible as well as for time saving issues; Moreover, an on-board reconstruction could be used as part of the control system, where the variations measured from consecutive images could help to understand the UAV recent trajectory. Furthermore, with the method proposed in this paper off-board panorama reconstruction from a flight video is also possible.

In this section the main steps of the algorithm are detailed, including explanation of how the maths behind each step work. Firstly, a matching between two preprocessed images is done to find common key aspects in successive frames, from this key aspects the homography matrix is determined. From the decomposed homography, rotation and translation are obtained. Thereafter, the image wrapping is used to overcome the UAV rotation and relative movement over the image. Next, with the obtained data the images are blended forming a panoramic image. This process is repeated using the area of interest of the resultant image and the following frame until the final panorama is created.

3.1 Image Matching

Both images are taken from the same camera in two different reference systems denominated: the first frame and the second frame. After this, the matching points are computed between the panoramic image and the next image.

The first step is the matching, a set of feature points are detected in each image, this features are image points denominated keypoints. For the set of keypoints, a set of vectors are defined mathematically that interprets the features of each keypoint. This vectors are designated as descriptors and are used to find equivalences between images. The correct equivalences are treated as inliers.

In this work, SURF algorithm is used to detect the keypoints and extract the descriptors for two reasons: it is a robust method as shown in [23], and from the methods used this the one gave the best results achieving a greater number of inliers in the shortest time.

The matching algorithm implemented is based on Fast Library for Approximate Nearest Neighbors (FLANN) [24]. This finds the two best matches for each descriptor, and sorts them by distance. This distance is then used to determine whether the match is suitable or not. From the valid matches, the homography matrix is then estimated.

3.2 Homography Matrix

To understand how the homography works, it is important to explain the notation used in this paper. The maths used calculating the homography are based on the methods seen in [25].

In Figure 1, the two frames in front the planar object are noted, where the second frame has been rotated and translated with respect to the first one, and where the first coordinate system axes are configured in an ideal angle. The homogeneous transformation matrix that allows to convert a 3D vector from the first to the second frame is:

$$ {}^{1}T_{2} = \begin{bmatrix} {}^{1}R_{2} & {}^{1}t_{2} \\ 0 & 1 \end{bmatrix} \quad (1) $$

This matrix is composed of the 3-by-3 rotation matrix ${}^{1}R_{2}$ and the 3D translation vector ${}^{1}t_{2}$. The two frames have the same optical center when the translation vector is the zero vector, on the other hand, when the rotation matrix is an identity matrix, the axes of the two frames would be parallel to each other.

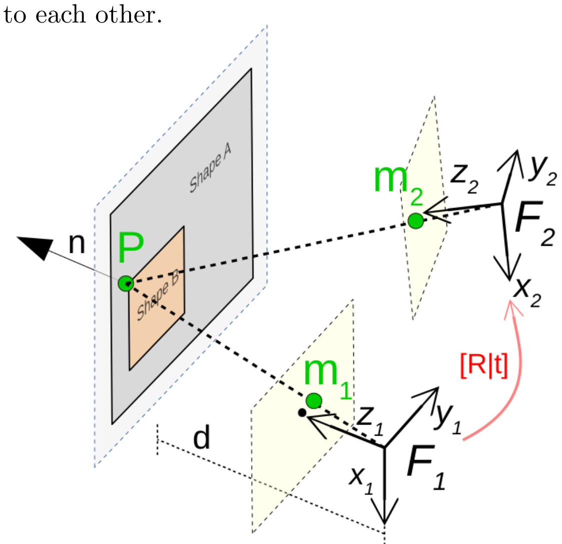

Figure 1: Point representation in two frames

The denotation of a 3D point within the planar object referred to the first and the second frame is P_1 and P_2 respectively. This point is normalized as follows:

$$ m_1 = \left(\frac{x_1}{z_1}, \frac{y_1}{z_1}, \frac{z_1}{z_1}\right) = (x_1{}', y_1{}', 1) \quad (2) $$

$$ m_2 = \left(\frac{x_2}{z_2}, \frac{y_2}{z_2}, \frac{z_2}{z_2}\right) = (x_2{}', y_2{}', 1) \quad (3) $$

This normalized coordinates are used to obtain the image coordinates in pixel using the following transformation:

$$ p = Km \quad (4) $$

where $k = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ is the intrinsic parameters matrix which is calculated in the camera calibration, where, $(f_x,\ f_y)$ are the camera focal lengths and $(o_x,\ o_y)$ are the optical center. The transforming from the normalized to the image coordinates is done using the following equations:

$$ u_1 = f_x \cdot x_1' + o_x \quad (5) $$

$$ v_1 = f_y \cdot y_1' + o_x \quad (6) $$

As a result of the operations, $(u_1, v_1, 1)$ are the vector coordinates of a pixel in the first image, and $(u_2, v_2, 1)$ are the vector coordinates in the second image, each of the pixels is relative to the top left corner. The following equation is used to convert the coordinate vector of the second image p_2 to the coordinate vector of the first image p_1:

$$s \cdot p_1 \quad = \quad \gamma k H k^{-1} p_2 \tag{7}$$

where s is a scale factor and H is the Euclidean homography matrix:

$$H \quad = \quad {}^1R_2 + \frac{{}^1t_2}{d_2} n^T \tag{8}$$

3.3 Image Warping

To eliminate the noise produced by the rotation of the camera, the translation matrix is considered as the zero vector to transform the second image points. Each point in the second image is remapped, using a linear interpolation, with the Equation:

$$p_2'' \quad = \quad s \cdot p_2' \tag{9}$$

where,

$$p_2' \quad = \quad k({}^1R_2)k^{-1}p_2 \tag{10}$$

From Equation 10, the points are transformed if the second frame axes were parallel to the first frame axes, but the image shapes does not have the same scale. Therefore, before transforming all the points, it is necessary to obtain the scale factor s of this transformation.

To get s, first of all, two pairs of matched points are selected (the two furthest). Secondly, the two inliers points of the second image are transformed using the Equation 10. Finally, the distances between the two first image inliers and the transformed inliers are measured and compared as follows:

$$s \quad = \quad \frac{l_1}{l_2} \tag{11}$$

where l_1 is the distance between two furthest inliers points in the first image and l_2 is the distance referred to the second image but with the points transformed.

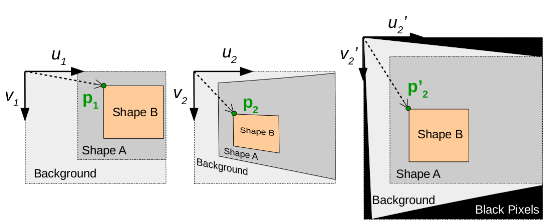

Figure 2: Image warping; a) Observed shapes from the first frame, b) Observed shapes from the second frame, and c) Second image converted by the Equation 9

3.4 Exposure Compensation

It is important to adjust the exposure between the two overlapping images before finding the seams; because although both images have common features, the exposure difference between both could be high and the brightness difference is noticed in whole panorama. Furthermore, the adjustment could help the seams search because a large exposure difference confuses the algorithm.

This technique is based on the division of an input image in different blocks, see [26]. Each block is compared in the luminescence sense with the image which is overlaid by the block. This method computes a quadratic transfer function for each block. Thereafter, the transfer functions of the neighboring blocks are averaged to smooth the variation of the functions distribution. Finally, the result of each pixel is obtained through a linear interpolation from neighboring patches of the input image.

In this process, a Region of Interest (ROI) is used as the first frame in order to reduce the computational time, this is explained in further detail in section 4.

3.5 Seam Finder

As a seam finder between the two images, a method based on the graph cut problem [27] is used. For the blending, the minimum graph cutting cost is calculated and applied in the overlapping region.

To apply the graph cut algorithm, first of all, each pixel within the overlapped area is defined as a node. The first image and second image are the patch A and B respectively. After, the adjacent pixels are connected with a arc. Each arc is labeled with the equation $M(s, t, A, B)$ called matching quality cost:

$$M \quad = \quad \|A(s) - B(s)\| + \|A(t) - B(t)\| \tag{12}$$

where s and t are the position of two adjacent nodes and where A and B are the pixel color in the different patches. For instance, $B(s)$ is the s pixel color in the B patch.

Finally, the arcs connected to a pixel outside the overlapping region are labeled with an infinitely high cost. Then, the graph cut problem is solved by drawing the path dividing both patches. This can be seen in Figure 3, where the red line represents the path, resulting in the pixels from the left area being copied from the patch A and the ones in the right from the B.

3.6 Feather Blending

For the blending of the two frames, a Feather-blending method is used; Originally multiband-

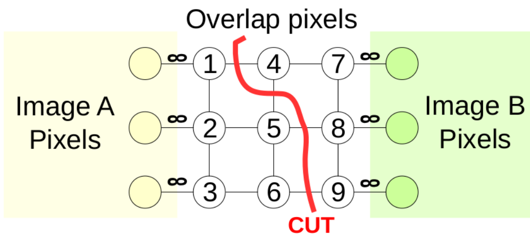

Figure 3: Path found by the graph cut algorithm

blending was used, however in the case presented it was identified that the solution obtained with feather-blending was more optimal as the Gaussian filter implemented in the original approach blurred the image. The method is based on a variation of weights given to the pixels of each image depending the distances from the edges [28].

To achieve the stitched image, it is necessary to know the input image positions in the output image. To do that, the image corners should be computed. The corner constitutes the top left point of each image and in the case of the first image it is the vector $(0, 0)$ and in the second image it is calculated relatively to the first image corner. The second image corner is obtained following a set of steps. First at all, warp the corner of the second image using Equation 10 and a inlier point, getting the warped points c_2'' and i_2''. After, determine the difference between a first image inlier coordinates and the corresponding warped inlier coordinates of the second image, getting the translation vector (t_x, t_y) in pixels. The components of the resulting corner are the sum of the warped corner plus the obtained translation vector.

Eventually, with the corners of the images that are stitched and the size in pixels of them, it is possible used the feather-blending method to obtain the panoramic image.

4 Experimental results

In this section, the real flight tests are performed in order to evaluate the proposed algorithm. In addition, the panorama for the initial tests is created from aerial images dataset taken from [29].

The initial test seek to find the optimal overlapping percentage between consecutive frames, so various datasets where tested at 50% and 80% overlap using the time and the quality of the image to evaluate the results. For the dataset of a neighborhood, seen in Figure 5 a panoramic at 80% overlap is composed using a total of 10 frames, as well as a panoramic at 50% overlap with 4 frames; it is noteworthy the fact that the same area is being covered in all tests. Additionally, a large panoramic image is shown from the stitching of 15 aerial frames, Figure 4.

The illumination compensation is also analyzed to determine the optimal approach to form the mosaic: compensating the changes in illumination over the whole image, over the section being analyzed or not including an illumination compensation at all.

Moreover, in the experiment done, a comparison of the computational time for the different sections of the algorithm is evaluated, analyzing the benefits of including the ROI and the exposure compensation.

4.1 Platform

In the experiments, DJI F450 quadcopter based on Pixhawk control system was used. The quadcopter is equipped with SJ4000 wireless camera that provides images of 640×480 pixels, mounted on Tarot G-2D gimbal to provide stability of the camera. The processing in the ground station is performed in Intel i5-2410M at 2.3 GHz CPU and 4 GB RAM. The connection with the UAV is established via a standard 802. 11n wireless LAN card.

4.2 Results

In this subsection, the results are shown and analyzed, Figure reffig:pano presents the final panoramic image created from aerial frames of a city with a 60% overlap.

From the analysis of the optimal overlapping percentage seen in Figure 5 it is observed that a 50% overlap gives a better result in the following aspects: a clearer image is obtained since fewer pixels are transformed every time the blending is applied, the panoramic image is created with reduced errors as there are less seams (see Figure 3), the computational time is reduced as the process is executed fewer times, and exposure compensation works better at lower overlapping area producing darker images at 80% overlap due to the compensation being applied more times. However, a possible drawback of using a low overlap as 50% would result in a lack of inliers found in images with a small number of key features. This would cause a failure in the mosaicking, nevertheless this has not been observed in the executed tests with the exception of the top-right corner of Figure 5 where the panoramic image presents the repetition of some features of the image due to the lack of matching features between consecutive frames. For this reason, it is concluded that the optimal overlapping percentage is slightly higher than 50%, so a 60% overlapping is used in the demonstration of the algorithm seen in Figure 4.

The exposure compensation for the panorama produces better results with large caption areas, i.e. when there is a considerable variation in illumi-

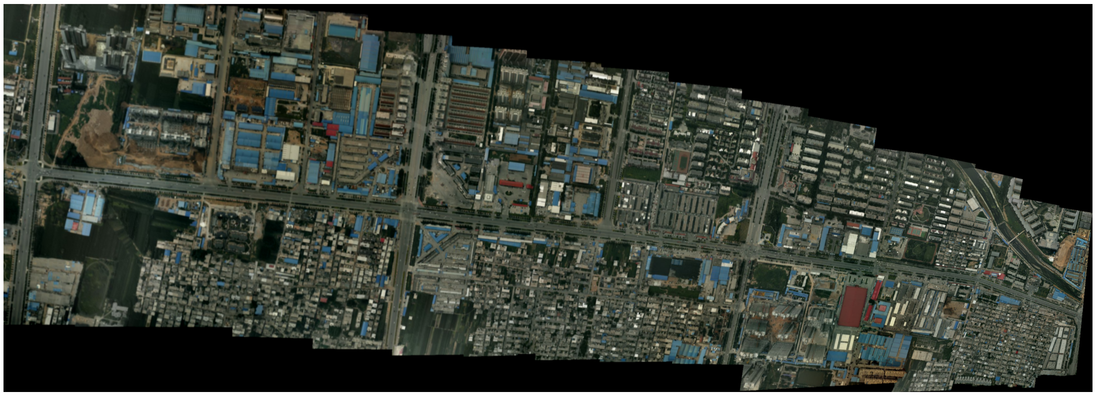

Figure 4: Mosaicking from aerial images

nation between some of the frames and others. Hence, the exposure compensation is beneficial for panoramas with large area covered, whereas in cases where the area covered is reduced, greater results are obtained without applying this compensation.

The use of the ROI improves the computational time of the algorithm without influencing the quality of the image. From Table 1 it is observed how the total computational time is reduced from 1.0245s to 0.9275s simply by the addition of a ROI.

Table 1 shows the computational time for the different sections of the algorithm. The times shown in this table represent the time taken for each individual section in the stitching of two frames. As the panoramic image grows, the times remain constant fluctuating depending on the number of keypoints detected in the pair of frames. These results are taken from the second stitching of the image in Figure 5b.

Table 1: Table showing times for each process in the image blending

Panoramic Steps	Time (s)
Detection & Matching	0.5246
Image warping	0.0302
Exposure compensation	0.0572 (0.1542)
Seam finder	0.2860
Blending	0.0287
Total	0.9275 (1.0245)

Note: numbers in parenthesis represent the occasions when the ROI is not applied.

5 Conclusion

In conclusion, this paper presents an algorithm capable of generating a panorama from aerial images, it can be observed the good set of results. Moreover, it is exposed from the results the benefits of including a ROI when compensating the illumination in the frames, as well as the fact that this compensation makes an effect when the panorama covers a large area. Furthermore, for the formation of the image mosaic, it is beneficial to use an overlapping of 60% since the computational time saved is significant as well as creating superior images.

During the experiment there were a series of problems encountered, some of which have been resolved and some are analyzed to solve in future works:

- The number of keypoints detected in smooth/dark surfaces created ghosting, which meant that some objects were missing from the final panorama.
- Multi-bland blending uses Gaussian filters over the whole panorama, blurring the image, hence Feather blending is used instead.
- The first image must be parallel to the structure being analyzed since, a failure to do this creates accumulated error as every image takes the coordinate system of the original.
- The concentration of keypoints on a specific area means that the homography produced is incorrect due to lack of information in the frames, hence keypoints must be scattered around the image.

From the problems encountered, it is determined that the use of Lucas-Kanade tracking for keypoint matching could potentially improve the results obtained from the proposed algorithm. Henceforth the next step to improve the algorithm would be to include the keypoint tracking, which would solve the issues experienced during the initial tests.

Moreover, as mentioned before, the original goal is to mount the camera on a UAV and to reconstruct the mosaic in real-time. Once, the code has been improved this would be a feasible task as it is a

(a) Panoramic at 60% overlap (b) Panoramic at 80% overlap

Figure 5: Image mosaic of a neighborhood at different overlapping percentages

fast and robust algorithm capable of running in real time.

Acknowledgments

Research supported by the Spanish Government through the CICYT projects (TRA2015-63708-R and TRA2016-78886-C3-1-R), and the Comunidad de Madrid through SEGVAUTO-TRIES (S2013/MIT-2713).

References

[1] G.-J. M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, and others, "Rescue robots at earthquake-hit mirandola, italy: a field report," in *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1–8.

[2] D. Erdos, A. Erdos, and S. Watkins, "An experimental UAV system for search and rescue challenge," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, May 2013.

[3] S. Choi and E. Kim, "Image Acquisition System for Construction Inspection Based on Small Unmanned Aerial Vehicle," in *Advanced Multimedia and Ubiquitous Engineering*. Springer, 2015, pp. 273–280.

[4] A. Al-Kaff, F. M. Moreno, L. J. San José, F. Garcìa, D. Martín, A. de la Escalera, A. Nieva, and J. L. Meana Garcìa, "VBII-UAV: Vision-based infrastructure inspection-uav," in *World Conference on Information Systems and Technologies*. Springer, 2017.

[5] J. Xu, G. Solmaz, R. Rahmatizadeh, D. Turgut, and L. Boloni, "Animal monitoring with unmanned aerial vehicle-aided wireless sensor networks," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*. IEEE, 2015, pp. 125–132.

[6] D. Anthony, S. Elbaum, A. Lorenz, and C. Detweiler, "On crop height estimation with UAVs," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4805–4812.

[7] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Auton Robot*, vol. 30, no. 1, pp. 73–86, Jan. 2011.

[8] L. Ma, M. Li, L. Tong, Y. Wang, and L. Cheng, "Using unmanned aerial vehicle for remote sensing application," in *Geoinformatics (GEOINFORMATICS), 2013 21st International Conference on*. IEEE, 2013, pp. 1–5.

[9] W. Tampubolon and W. Reinhardt, "UAV Data Processing for Large Scale Topographical Mapping," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5, pp. 565–572, Jun. 2014.

[10] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta, "Autonomous searching and tracking of a river using an UAV," in *American Control Conference, 2007. ACC'07.* IEEE, 2007, pp. 359–364.

[11] C. Yuan, Z. Liu, and Y. Zhang, "UAV-based forest fire detection and tracking using image processing techniques," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on.* IEEE, 2015, pp. 639–643.

[12] L. T. Lilien, L. b. Othmane, P. Angin, B. Bhargava, R. M. Salih, and A. DeCarlo, "Impact of Initial Target Position on Performance of UAV Surveillance Using Opportunistic Resource Utilization Networks." IEEE, Sep. 2015, pp. 57–61.

[13] S. Pravenaa and R. Menaka, "A methodical review on image stitching and video stitching techniques," *International Journal of Applied Engineering Research*, vol. 11, no. 5, pp. 3442–3448, 2016.

[14] M. Alomran and D. Chai, "Feature-based panoramic image stitching," in *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on.* IEEE, 2016, pp. 1–6.

[15] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[16] F. E. Sandnes and Y.-P. Huang, "Translating the viewing position in single equirectangular panoramic images," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on.* IEEE, 2016, pp. 000 389–000 394.

[17] R. B. Inampudi, "Image mosaicing," in *Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS'98. 1998 IEEE International*, vol. 5. IEEE, 1998, pp. 2363–2365.

[18] D. Wei and G. Zhou, "Real-time uav ortho video generation," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 5. IEEE, 2008, pp. V–510.

[19] X. Yin and N. Ma, "Study on the automatic unmanned aerial vehicle image mosaic algorithm," in *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, vol. 2. IEEE, 2011, pp. 624–628.

[20] H. Wang, J. Li, L. Wang, H. Guan, and Z. Geng, "Automated mosaicking of uav images based on sfm method," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International.* IEEE, 2014, pp. 2633–2636.

[21] H. Meuel, S. Ferenz, M. Munderloh, H. Ackermann, and J. Ostermann, "In-loop radial distortion compensation for long-term mosaicing of aerial videos," in *Image Processing (ICIP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 2961–2965.

[22] C. Eschmann, C. Kuo, C. Kuo, and C. Boller, "Unmanned aircraft systems for remote building inspection and monitoring," in *6th European workshop on structural health monitoring*, 2012, pp. 1–8.

[23] Z. Pusztai and L. Hajder, "Quantitative comparison of feature matchers implemented in opencv3," 2016.

[24] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration." *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[25] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," Ph.D. dissertation, INRIA, 2007.

[26] M. Uyttendaele, A. Eden, and R. Skeliski, "Eliminating ghosting and exposure artifacts in image mosaics," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–II.

[27] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics (ToG)*, vol. 22, no. 3. ACM, 2003, pp. 277–286.

[28] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics (TOG)*, vol. 2, no. 4, pp. 217–236, 1983.

[29] M. Xia, J. Yao, R. Xie, L. Li, and W. Zhang, "Globally consistent alignment for planar mosaicking via topology analysis," *Pattern Recognition*, vol. 66, pp. 239–252, 2017.