Fast Image Mosaicing using Incremental Bags of Binary Words

Emilio Garcia-Fidalgo, Alberto Ortiz, Francisco Bonnin-Pascual and Joan P. Company

Abstract— **Image mosaicing has gained increasing attention in the last few years, specially for robotic mapping applications. Due to the richness of the sensor data provided, several science fields require the creation of large-area image mosaics for further analysis. In this paper, we propose a novel and generic image mosaicing approach that can produce seamless composites under different configurations in a reasonable amount of time. Our approach is based on a multi-threaded architecture which allows us to execute the different steps of the algorithm simultaneously. To find the topology of the environment, we use a visual index based on a Bag-of-Binary-Words scheme, which is built in an online manner, and thus avoids the classic training step. Our approach is validated under different environments and camera configurations, showing that it can be used on several scenarios producing coherent results in all of them. Furthermore, the implementation of the algorithm is made public to the community.**

I. Introduction

Robots are becoming more important for automating tasks, specially in places that present a difficult access for humans. In this regard, several platforms have been recently proposed for vessel inspection [1], [2], underwater surveying [3]–[8] or aerial mapping [9]–[11], where the operating conditions can be dangerous for human intervention. In the last decades, cameras have been widely used for collecting information from the environment, due to their low cost, the richness of the sensor data provided and the availability of cheap powerful computers. When a robot is equipped with a camera, it is usually of interest to obtain a large visual representation of the operating area, which can be used for close-up inspection, for localization and even for navigation tasks. Since the field-of-view of conventional cameras is limited, image mosaicing techniques have been developed for building a larger view of the surveyed area. Mosaicing is then defined as the process of stitching images together to provide a wide-area image of the scene.

One of the key steps for image mosaicing is the estimation of the topology. The quality and the time needed to obtain the final topology are directly related to the method used for describing images and the ability for finding overlapping pairs. With regard to image description, most part of the existent image mosaicing approaches make use of SIFT [12] or SURF [13], due to their invariance properties to illumination, scale and rotation changes. However, recently there has been a growing interest in the development of binary descriptors, such as BRIEF [14], ORB [15], BRISK [16] or LDB [17], which are faster to compute and require less storage space. In order to detect the existent relationships between images, if no other source of information is present, a frame-to-frame comparison approach can be used only when the number of images is low. As it grows, this approach becomes unfeasible and an indexing scheme is needed for searching overlapping pairs in an efficient way. The Bag-of-Words (BoW) approach [18], commonly used for image retrieval, is of application here. However, despite its good results, this technique presents several drawbacks that can affect the global performance of a mosaicing algorithm, such as the training step.

Image mosaicing has drawn attention of the robotics community some years ago, specially for mapping areas using down-looking cameras, as the most part of the approaches presented so far. However, it is less usual to find solutions that make use of forward-looking cameras, as presented in [3] for underwater environments and in [2] for a Micro-Aerial Vehicle (MAV). Furthermore, image mosaicing algorithms are usually validated to work in only one environment.

In this paper, we propose a novel image mosaicing approach, named BIMOS (Binary descriptor-based Image MOSaicing), which can produce seamless mosaics on different scenarios and camera configurations in a reasonable amount of time. More precisely, we introduce a multi-threaded architecture for image mosaicing that allows us to decouple the strategic steps involved in the mosaicing process, speeding up the time required to estimate the final topology. To find overlapping candidates, we employ a binary visual dictionary [19], which is based on a BoW scheme that is built in an online manner. Our approach takes advantage of the use of the ORB detector and descriptor [15] to accelerate the image description process.

In our previous work [2], we presented an image mosaicing approach to work with sequences of images captured by a MAV. The resulting mosaics were then used during a vessel inspection process. In this work, we go a step further presenting a more generic mosaicing approach based on a multi-threaded architecture and a new image selection policy, which makes the current solution even faster than our previous work, as will be shown in the results. Furthermore, we evaluate our algorithm in several environments to validate that it can be used in different scenarios and using several camera configurations. As an additional contribution, we release the code of BIMOS as a ROS (Robot Operating

This work is partially supported by the European Social Fund through grant FPI11-43123621R (Conselleria d'Educacio, Cultura i Universitats, Govern de les Illes Balears), and by projects MERBOTS (DPI2014-57746-C3-2-R) and INCASS. The project INCASS has received research funding from the EU FP7 under GA 605200. This publication reflects only the author's views and the European Union is not liable for any use that may be made of the information contained therein.

All authors are with the Department of Mathematics and Computer Science, University of the Balearic Islands, 07122 Palma de Mallorca, Spain, email: emilio.garcia@uib.es

System) node[1] so that other researchers can use it.

The rest of the paper is organized as follows: Section II describes our approach for indexing images, which is used for detecting overlaps between images, Sections III and IV describe our mosaicing approach, Section V reports the experimental results obtained, and Section VI concludes the paper.

II. Search for Overlapping Pairs

As mentioned above, finding the existent relationships between the images of the sequence is of prime importance for the correct estimation of the topology. If we want to avoid an image-to-image approach, we need a fast and efficient method to determine overlapping image pairs. Image retrieval methods developed recently are based on the BoW approach [18]. Despite its good performance, this technique presents several drawbacks, since it usually needs a training phase, and the generated visual dictionary can be non-representative for all environments. Furthermore, most BoW approaches for image indexing are usually based on real-valued descriptors [12], [13] and is less common to find binary solutions [20]. In this work, we employ a method for computing a vocabulary of binary features that can be built online, avoiding thus the training phase. This method, called OBIndex (Online Binary Index)[2], is used as a base in our approach to estimate the topology of the environment. The algorithm is briefly reviewed next for completeness. For further details, the reader is referred to [19].

Our method is built over on an incremental visual dictionary based on a modified version of Muja and Lowe's approach [21]. The dictionary is combined with an inverted index, which contains, for each visual word, a list of images where it was found.

Since our approach relies on an incremental visual dictionary based on binary features, an updating policy for combining binary descriptors is needed. Averaging each component of the vector is an option for real-valued descriptors, but it cannot be considered for the binary case. OBIndex uses a bitwise AND operation. Formally, being B a binary descriptor:

$$B_{w_i}^t = B_{w_i}^{t-1} \wedge B_q \,, \tag{1}$$

where $B_{w_i}^{t-1}$ is the binary descriptor of the word w_i stored in the dictionary at time $t-1$, B_q is the query descriptor and $B_{w_i}^t$ is the merged descriptor for word w_i at time t. This policy is inspired by the observation that each component of a binary descriptor is usually set to 1 or 0 according to the result of a comparison between a pair of image pixel intensities. If the *i*-th bit is the same in both descriptors, it means that the result of this comparison between the pixel intensities was the same in both images. Otherwise, we experimentally prioritize the use of the zero value by means of the AND operation.

The index is initially built using the descriptors of the first image as visual words. When a new image needs to be added to the index, their descriptors are searched in the index. Given a query binary descriptor, we search for the two nearest neighbours traversing the tree from the root to the leafs and selecting at each level the node that minimizes the Hamming distance. Using these two neighbours, we apply the ratio test [12] using a threshold of 0.8 to determine if both descriptors represent the same visual feature. If positive, the query descriptor and the visual word are merged using (1) and the latter is replaced in the dictionary. Otherwise, the query descriptor is considered a new feature and is added to the index as a new visual word. In both cases, the inverted index is updated accordingly, adding a reference to the current image in the list corresponding to the modified or added word. Given the features of a query image as input, OBIndex returns an ordered list of images according to a scoring process based on Term Frequency Inverse Document Frequency (TF-IDF) weighting [22].

III. Motion Estimation

Once two images have been detected as an overlapping pair, the alignment between them is determined. The model employed to estimate the image motion plays a key role in the image registration process. BIMOS assumes that either the scene is planar or the distance from the camera to the scene is high enough so as to neglect the depth changes. It is also assumed that the camera is more or less perpendicular to the scene and at a more or less constant distance.

Under these conditions, two overlapping images I_i and I_j are related by a homography, a linear transformation represented by a 3×3 matrix ${}^iH_j$ such that $p_i = {}^iH_j\, p_j$, where p_i and p_j are two corresponding points from, respectively, I_i and I_j, expressed in homogeneous coordinates. Despite BIMOS can deal with affine transformations (six degrees of freedom), we approximate the motion of the camera by a simpler model using a similarity transformation, which has four degrees of freedom comprising rotation, translation and scaling. Therefore, ${}^iH_j$ is expressed as:

$${}^iH_j = \begin{pmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

where s is the scale, θ the rotation angle and (t_x, t_y) the translation vector. The estimation of any of these homographies starts by matching corresponding points between images. Maximum Likelihood Estimation Sample Consensus (MLESAC) [23] is next used as a robust estimation algorithm to minimize the reprojection error for (2) and discard outliers. Finally, for the case of a path of images $I_i, I_{k_1}, \ldots, I_{k_m}, I_j$, the associated transformation that relates frames I_i and I_j can be computed by concatenating the corresponding relative homographies: ${}^iH_j = {}^iH_{k_1}\, {}^{k_1}H_{k_2} \ldots {}^{k_{m-1}}H_{k_m}\, {}^{k_m}H_j$.

IV. Image Mosaicing using Binary Descriptors

In this section, we describe BIMOS, whose architecture is outlined in Fig 1. Inspired by ORB-SLAM [24], the system consists of four threads that run in parallel, each one in charge of a strategic step of the algorithm. This configuration

[1] http://github.com/emiliofidalgo/bimos

[2] http://github.com/emiliofidalgo/obindex

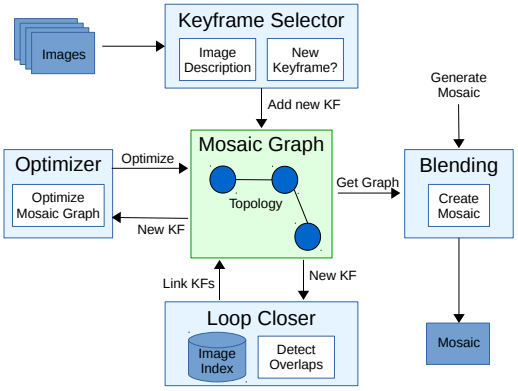

Fig. 1. BIMOS architecture. The four threads (in light blue) interact with a shared structure called *mosaic graph* (in light green). The arrows indicate the main actions performed between the different components. See text for further details.

allows us to decouple the execution of the different parts of BIMOS, reducing the time needed to generate a mosaic. All threads interact with a shared structure called *mosaic graph*, which is used to manage the topology of the environment and the synchronization mechanisms between threads. The *keyframe selector* thread, which is the entry to the system, describes the input images and decides if they should be part of the final composite. The *loop closer* thread detects overlapping image pairs and the *optimizer* thread reduces the global misalignment in the graph performing a bundle adjustment process. Finally, the *blending* thread is responsible for generating the final image mosaic. BIMOS is ready to work online using a ROS topic through which it processes images on demand, contrary to most mosaicing algorithms, which work offline. In the following sections, we describe the building blocks of BIMOS.

A. Mosaic Graph

The topology of the environment represents the relationships that exist between the images conforming the mosaic. In our approach, the topology is modelled by means of an undirected graph, where nodes represent a selected set of images that will be included in the final mosaic and links represent the overlaps between them. In BIMOS, the selected images are called *keyframes*.

The *mosaic graph* is a key component of BIMOS. It manages the graph that represents the topology of the environment and provides mechanisms to ensure the exclusive access of the different threads to this graph. In order to create the final mosaic, keyframes need to be aligned according to a common selected frame, referred to as the *mosaic frame*. Then, each keyframe is associated to an absolute homography ${}^{M}H_i$, which relates the correspondent keyframe i with the mosaic frame M. In our previous work [2], the mosaic frame was selected as the node with the highest output degree after the graph construction was completed. In this work, since BIMOS processes images on demand and the graph is updated as new images arrive, the first keyframe is always selected as the reference frame of the mosaic. Its absolute homography is thus the identity matrix. Each link

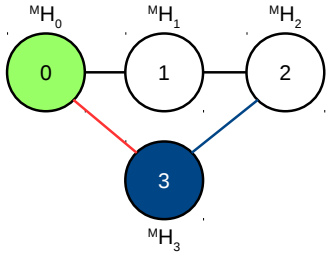

Fig. 2. Example of a mosaic graph comprising four keyframes with their corresponding absolute homographies. The keyframe 0, coloured in green, is the mosaic frame and, therefore, ${}^{M}H_0$ is the identity. The keyframe 3 is the last one inserted in the graph. The node and the link with the keyframe 2, marked in blue, were added by the *keyframe selector* thread. The link with the keyframe 0, marked in red, was added by the *loop closer* thread after detecting an overlap between the images. The *optimizer* thread will tune the absolute homographies of the graph.

is also associated to a relative homography, which will be used during the pose-graph optimization step.

Several threads of the system modify concurrently the mosaic graph: the *keyframe selector* thread inserts new keyframes in the graph, the *loop closer* thread links keyframes as it detects overlapping image pairs and the *optimizer* thread globally adjusts the absolute homographies ${}^{M}H_i$. The mosaic graph structure and its use are illustrated in Fig 2.

B. Keyframe Selection

This component is responsible for describing the input images and deciding which ones are useful for building the final mosaic. First of all, the ORB [15] algorithm is used to detect and describe a set of keypoints in the image. We use ORB due to its good tolerance to rotations [24], instead of FAST [25] and LDB [17] as in our previous solution [2]. However, note that BIMOS is descriptor-independent and any detector-descriptor combination including a binary descriptor can be used. Besides, to favour accurate estimation of the image transformations, a minimum number of features (3000) is requested to be found, and they are required to cover the full image in a more or less uniform way defining a 4×4 regular grid over the image.

Instead of using all the input images, we apply a keyframe selection policy in order to discard images which are not deemed to provide a significant contribution to the mosaic, avoiding unnecessary drift during the alignment process. This contribution is measured as the amount of overlap between the current image and the last keyframe inserted in the graph, so that the higher the overlap the less relevant is the image. More specifically, we compute the homography ${}^{k}H_i^*$ between the current image i and the last inserted keyframe k. Given the resulting set of inliers, we obtain the coordinates of the corresponding points in each image. We then calculate the minimal up-right bounding rectangle for each point set, formally r_j for image j, and, next, we evaluate the percentage of overlap that this bounding rectangle represents in the image. This overlap is expressed as follows:

$$O_j = \frac{\text{area}(r_j)}{w_j \times h_j}, \tag{3}$$

where the function area$(\cdot)$ computes the area of the bounding rectangle and w_j and h_j are, respectively, the width and the height of image j in pixels. In order to take a final decision, the overlap between the images is computed as:

$$ {}^{k}O_i = min(O_k, O_i)\,. \tag{4} $$

Then, if the number of inliers is higher than a threshold τ_{in} and ${}^{k}O_i$ is higher than another threshold τ_{ov}, the current image i is stored as a potential keyframe. Otherwise, the last potential keyframe found is added to the mosaic graph, and the transformation from the current image to the new keyframe is recomputed. This policy allows us to ensure that, despite we are discarding several images, there exists a minimum overlap between consecutive keyframes and the topology is not broken in different parts.

When an image i is added as keyframe into the mosaic graph, it is linked with the previous keyframe. The link is associated to the computed homography ${}^{k}H_i^*$ and the absolute homography is initialized concatenating the homography of the previous keyframe with ${}^{k}H_i^*$. Then, following the notation used in this paper, if the image i is added as the keyframe $k+1$ in the graph, ${}^{k}H_i^*$ becomes ${}^{k}H_{k+1}$ and, consequently, the initial absolute homography can be written as:

$$ {}^{M}H_{k+1} = {}^{M}H_k\, {}^{k}H_{k+1}\,. \tag{5} $$

C. Loop Closing

This thread detects which keyframes close a loop with previously added keyframes. To this end, we use our indexing scheme explained in Section II. This component maintains an instance of OBIndex, which indexes all the keyframes defined up to the current time. When a new keyframe is received, it is searched in the index, obtaining a list of candidates sorted from highest to lowest visual similarity. Next, each candidate is evaluated in descending order, computing the homography with the current keyframe. If the number of resulting inliers is higher than a certain threshold, a link between the corresponding keyframes is incorporated into the graph. Otherwise, the process finishes and, if exists, next keyframe is processed.

Since consecutive images are linked by default, we want to find overlapping pairs at farther distances, which is of prime importance during the optimization step. To achieve this, keyframes are not directly indexed as soon as they are processed. Instead, a buffer is used to store the most recent keyframes, delaying their publication as overlapping candidates for the following processed keyframes.

D. Optimization

Despite the efforts for accurately estimating the topology, alignment errors still arise, resulting into globally inconsistent mosaics. To correct this problem, this component is in charge of performing a bundle adjustment step to jointly minimize the global misalignment induced by the current absolute homographies. The error function is defined as follows:

$$ \begin{aligned} \epsilon = \sum_i \sum_j \sum_{k=1}^{n} & \|p_i^k - ({}^{M}H_i)^{-1}\, {}^{M}H_j\, p_j^k\| \;+\; R({}^{M}H_j) \\ & \|p_j^k - ({}^{M}H_j)^{-1}\, {}^{M}H_i\, p_i^k\| \;+\; R({}^{M}H_i)\,, \end{aligned} \tag{6} $$

where i and j are two images related by a link, n is the total number of resulting inliers when computing the related homography, (p_i^k, p_j^k) are the corresponding points for the inlier k, ${}^{M}H_i$ and ${}^{M}H_j$ are the absolute homographies for, respectively, images i and j, and $R({}^{M}H_i)$ and $R({}^{M}H_j)$ are regularization terms. These terms prioritize homographies with scale closer to 1 during the optimization, since BIMOS assumes that the camera moves at a more or less constant distance from the scene, and are defined as follows:

$$ R({}^{M}H_i) = \gamma\left(a^2 + b^2 - 1\right) = \gamma\left((s\,\cos\theta)^2 + (s\,\sin\theta)^2 - 1\right) \tag{7} $$

where γ is a regularization factor, s and θ are the, respectively, scale and orientation contained in the homography, and a and b are defined in (2). To reduce the influence of outliers, we optimize, instead of (6), a Huber robust error function $h(\epsilon) = \{|\epsilon|^2 \text{ if } |\epsilon| \leq 1; 2|\epsilon| - 1 \text{ if } |\epsilon| > 1\}$. The system of non-linear equations is solved by means of the Levenberg-Marquardt algorithm using the Ceres Solver library[3] and the absolute homographies available so far as a starting point. Usually a few iterations are needed to achieve convergence.

Differently to our previous solution [2], where the bundle adjustment step was executed once after the topology estimation phase, in this work a short optimization is executed periodically after the insertion of a certain number of keyframes in the graph, limiting the optimization to a maximum of 30 seconds and 50 iterations. This parameter is of prime importance in the performance of the algorithm, since excessive optimizations may slow down the process. Just before the blending step, a longer optimization (a maximum of 600 seconds and 1000 iterations) is also performed to finally adjust the absolute homographies. Note that, despite the different convergence criteria, both optimizations adjust the absolute homographies in the whole graph. Instead, a local optimization could be performed only taking into account the part of the graph involving a detected loop closure. The implementation of this feature is proposed as future work.

E. Blending

This last component makes use of the multi-band blending algorithm [26] to create the final seamless mosaic. As in [2], this step is an adaptation of the *stitching* module implemented in the OpenCV library, which includes seam finding and exposure compensation. In BIMOS, this component runs as a thread on demand, which permits generating mosaics at different moments along the process.

[3] http://ceres-solver.org/

TABLE I

SUMMARY OF THE EXPERIMENTAL RESULTS. TIMES ARE EXPRESSED IN SECONDS AND ERRORS ARE EXPRESSED IN PIXELS.

			BIMOS							Approach [2]		
				Ex. Times				Rep. Error		Ex. Time	Rep. Error	
Seq	Size	#Imgs	KFs	Alig	Opt	Blend	**Total**	**Avg**	Std	**Total**	**Avg**	Std
VALLDEMOSSA1	320×180	201	80	14.44	0.42	38.71	53.57	2.25	2.27	187.52	2.71	3.02
VALLDEMOSSA2	1024×768	2504	335	330.07	0.44	2876.21	3206.72	8.08	14.75	9042.95	7.94	10.21
MAV	752×480	137	88	3.69	0.16	21.48	25.33	1.84	1.76	108.64	2.15	2.11
AIR1	800×533	71	32	5.34	0.50	68.65	74.49	4.29	6.61	224.39	4.12	3.24
AIR2	800×533	840	336	106.92	2.66	1008.43	1118.01	6.29	6.76	4766.64	5.94	5.31

V. EXPERIMENTAL RESULTS

We have validated our approach under different operating conditions using several datasets. The results obtained for each dataset are summarized in Table I, indicating the size of the images comprising the dataset (Size), the total number of images in the input set (#Imgs), the number of keyframes selected by BIMOS (KFs), the execution times corresponding to the different phases of the algorithm —global alignment (Alig), global optimization (Opt), blending (Blend) and the total time needed to build the mosaic (Total)— and, finally, the average and standard deviation of the reprojection error calculated using all the correspondences with the resulting set of homographies (Avg, Std). Note that the global alignment time also includes the small optimizations produced during the estimation of the topology. We also include in the table the execution time and the reprojection error of our previous image mosaicing algorithm [2] in order to show the performance improvement that BIMOS presents against this solution. All experiments were performed on a desktop computer fitted with an Intel Core i7 at 4.4Ghz processor and 32GB of RAM memory.

As a first experiment, we use an underwater dataset whose images come from the Valldemossa harbour seabed (Mallorca, Spain) and a hand-held down-looking camera. The dataset consists of 201 images of 320×180 pixels, which comprises a large loop, what allows us to validate the ability of our algorithm for recognizing previously seen places. A total number of 80 images were selected by BIMOS, leading to the final mosaic and the topology shown in Fig 3. Despite the reprojection error is similar to one obtained with our previous approach, the time needed to complete the mosaic is only 53.57 seconds in front of 187.52, which implies an increase of performance of 3.5x regarding the execution time.

The second dataset, also recorded at Valldemossa harbour, is a more challenging environment in the sense that it comprises a *Posidonia* meadow, characterized by a self-similar texture and vegetation in continuous motion. A total number of 2504 images were obtained, covering an area of approximately 400 m^2. BIMOS selects 335 as keyframes, producing the mosaic and the topology shown in Fig 4. As in the previous experiment, we obtain a coherent mosaic in less time than our previous approach.

The third dataset was recorded using a MAV designed for vessel visual inspection [27], which was fitted with a 752×480-pixel/58^o-lens uEye UI-1221LE camera running at 10Hz. Since currently we do not have access to a real vessel, we created a canvas of size 2.5×4 meters using

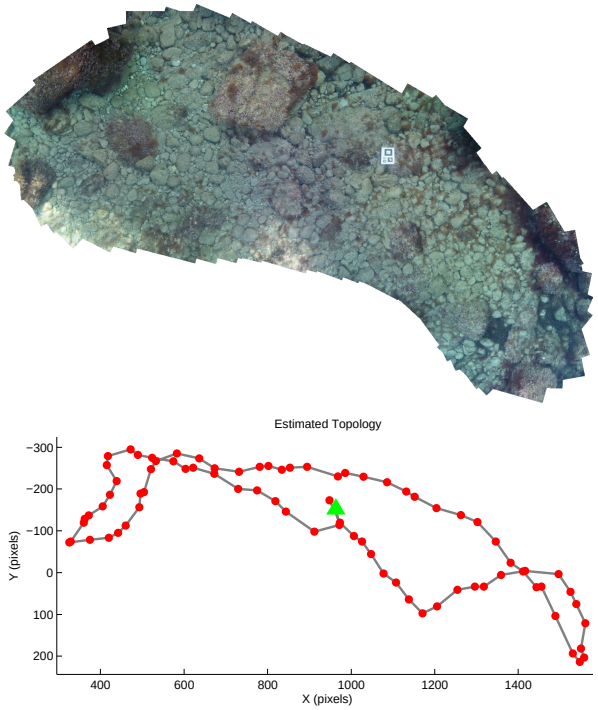

Fig. 3. (top) Resulting mosaic obtained for the Valldemossa1 dataset. (bottom) Topology estimated by BIMOS. Each keyframe is indicated using a red circle, and the mosaic frame is labelled by a green triangle.

images from a real cargo hold of a container ship. Next, we flew in front of the canvas with the MAV at a more or less constant distance from the wall (1.5m), performing a top down trajectory at the central part of the canvas. Note that due to the more aggressive dynamics of the MAV and the front-looking camera configuration, this is a more challenging situation than the Valldemossa dataset. A total number of 137 images were captured, from where BIMOS selects 88 as keyframes. As in the previous experiment, the reprojection error is lower than for our previous approach. However, the most interesting result has to do with the execution time of BIMOS, which is, according to our results, 4.3 times faster than our previous algorithm. The resulting mosaic and the estimated topology are shown in Fig. 5.

As a fourth experiment, we employ an aerial image sequence taken at a high altitude. This dataset was taken using a bottom-looking camera attached to an aerial vehicle [11], which was controlled manually by an operator. Note that this is a challenging scenario since the movement of the vehicle is more aggressive than in the previous cases, what makes the camera be far from perpendicular to the scene sometimes. We have considered two sequences from this

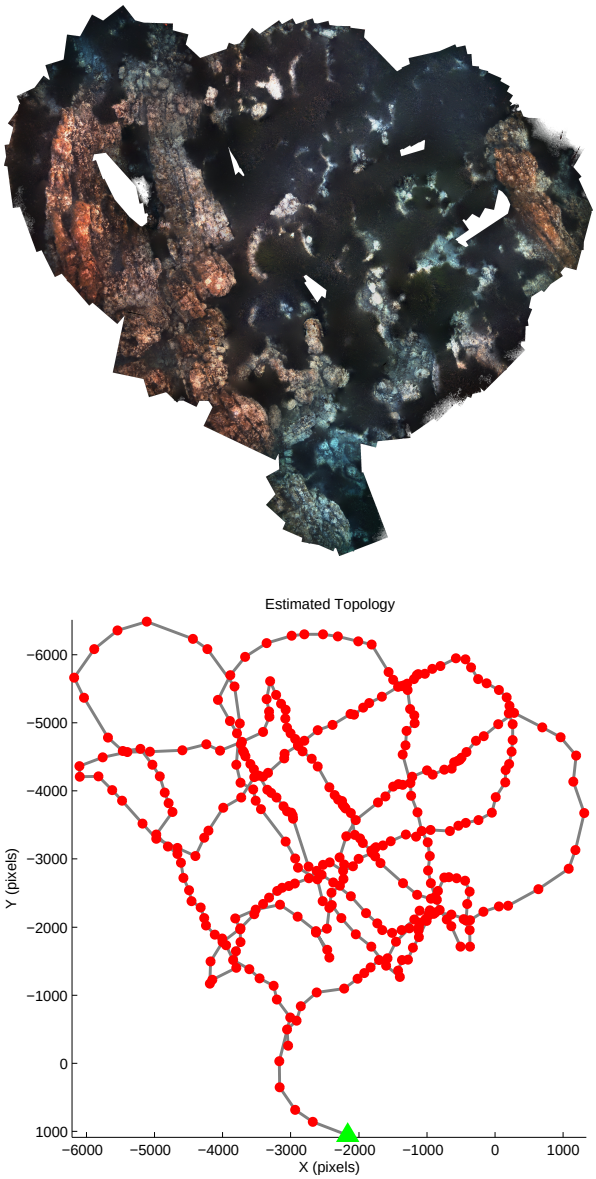

Fig. 4. (top) Resulting mosaic obtained for the Valldemossa2 dataset. (bottom) Topology estimated by BIMOS. Each keyframe is indicated using a red circle, and the mosaic frame is labelled by a green triangle.

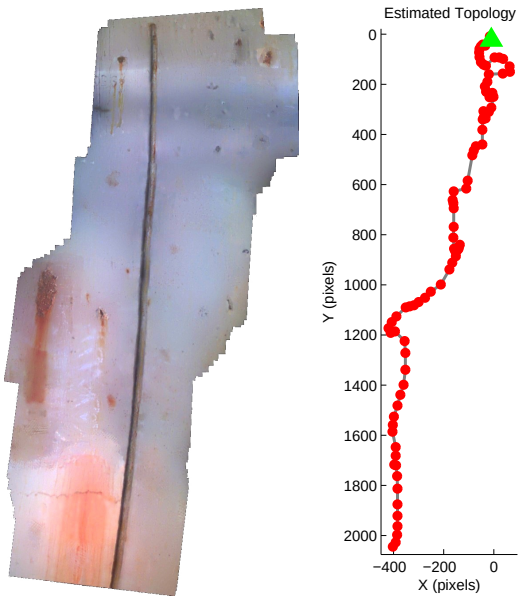

Fig. 5. (left) Resulting mosaic using the images collected by a MAV. (right) Topology estimated by BIMOS. Each keyframe is indicated using a red circle, and the mosaic frame is labelled by a green triangle.

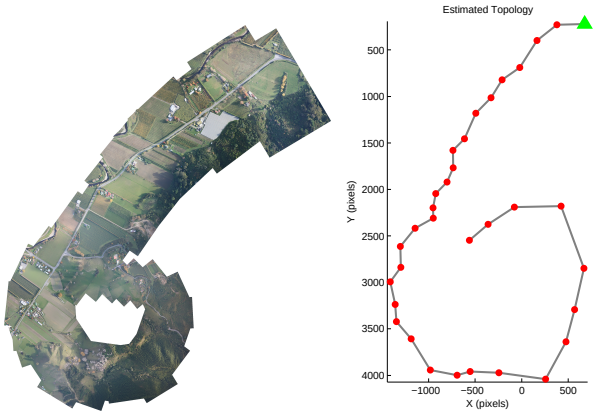

Fig. 6. (left) Resulting mosaic using the *AIR1* sequence. (right) Topology estimated by BIMOS. Each keyframe is indicated using a red circle, and the mosaic frame is labelled by a green triangle.

dataset, comprising, respectively, 71 and 840 images of size 800×533 pixels, corresponding to areas covering several kilometres. Each sequence is identified in Table I as *AIR1* and *AIR2*. The corresponding mosaics and the estimated topologies are shown, respectively, in Fig 6. and Fig 7. As in the other experiments, BIMOS is faster than our previous approach keeping a similar reprojection error.

In general terms, BIMOS is faster than our previous solution, but still producing coherent mosaics and maintaining a similar performance according to the reprojection error measure. The multi-threaded architecture and the keyframe selection policy can be considered as the main reasons to this increment of speed. Typically, most of the BIMOS execution time is invested at the blending step, an external module adapted to be used in BIMOS, whose seam finding algorithm is computationally demanding. We plan to make our own implementation of this step of the algorithm to further improve the execution times of BIMOS.

VI. Conclusions

In this paper, we have described a novel approach for generating mosaics from images. Our scheme, named BIMOS, is based on a multi-threaded architecture which allows us to decouple the different parts of the algorithm, speeding up the mosaicing process. The topology of the environment is modelled by means of an undirected graph. To find the overlapping pairs in an efficient way, this graph is created using a visual index of binary features, which is built online. We have validated our approach under different operating conditions, obtaining coherent mosaics in all cases. As a secondary contribution, the code of BIMOS has been made public to the community as a ROS node.

As a future work, as said previously, we are interested in improving the blending component of the algorithm, since

Fig. 7. (top) Resulting mosaic using the *AIR2* sequence. (bottom) Topology estimated by BIMOS. Each keyframe is indicated using a red circle, and the mosaic frame is labelled by a green triangle.

it is an external code and is the main bottleneck of our current solution. To further speed up the process, we plan to adopt a local optimization strategy instead of adjusting the whole graph during the global alignment process. We are also considering to extend the approach to use other motion models and/or multiple camera configurations. Finally, BIMOS can be further improved by merging image information with navigation data obtained from other sensors.

ACKNOWLEDGMENT

The authors would like to thank Tom Botterill and Francisco Bonin-Font for providing us, respectively, the aerial and the Valldemossa datasets used in this work.

REFERENCES

[1] M. Eich, F. Bonnin-Pascual, E. Garcia-Fidalgo, A. Ortiz, G. Bruzzone, Y. Koveos, and F. Kirchner, "A Robot Application for Marine Vessel Inspection," *J. Field Rob.*, vol. 31, no. 2, pp. 319–341, 2014.

[2] E. Garcia-Fidalgo, A. Ortiz, F. Bonnin-Pascual, and J. P. Company, "A Mosaicing Approach for Vessel Visual Inspection using a Micro Aerial Vehicle," in *IROS*, 2015.

[3] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, "Visual Inspection of Hydroelectric Dams using an Autonomous Underwater Vehicle," *J. Field Rob.*, vol. 27, no. 6, pp. 759–778, 2010.

[4] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Mosaic-Based Navigation for Autonomous Underwater Vehicles," *IEEE J. Oceanic Eng.*, vol. 28, no. 4, pp. 609–624, 2003.

[5] O. Pizarro and H. Singh, "Toward Large-Area Mosaicing for Underwater Scientific Applications," *IEEE J. Oceanic Eng.*, vol. 28, no. 4, pp. 651–672, 2003.

[6] H. Madjidi and S. Negahdaripour, "Global Alignment of Sensor Positions with Noisy Motion Measurements," *IEEE Trans. Rob.*, vol. 21, no. 6, pp. 1092–1104, 2005.

[7] A. Elibol, R. Garcia, and N. Gracias, "A New Global Alignment Approach for Underwater Optical Mapping," *Ocean Eng.*, vol. 38, no. 10, pp. 1207–1219, 2011.

[8] F. Ferreira, G. Veruggio, M. Caccia, E. Zereik, and G. Bruzzone, "A Real-Time Mosaicking Algorithm Using Binary Features for ROVs," in *MED*, 2013, pp. 1267–1273.

[9] T. Kekec, A. Yildirim, and M. Unel, "A New Approach to Real-Time Mosaicing of Aerial Images," *Rob. Auton. Syst.*, vol. 62, no. 12, pp. 1755–1767, 2014.

[10] H. Bulow and A. Birk, "Fast and Robust Photomapping with an Unmanned Aerial Vehicle (UAV)," in *IROS*, 2009, pp. 3368–3373.

[11] T. Botterill, S. Mills, and R. Green, "Real-Time Aerial Image Mosaicing," in *IVCNZ*, 2010, pp. 1–8.

[12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *ECCV*, vol. 3951, 2006, pp. 404–417.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF : Binary Robust Independent Elementary Features," in *ECCV*, 2010, pp. 778–792.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *ICCV*, 2011, pp. 2564–2571.

[16] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *ICCV*, 2011, pp. 2548–2555.

[17] X. Yang and K.-T. Cheng, "Local Difference Binary for Ultrafast and Distinctive Feature Description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 188–94, 2014.

[18] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *ICCV*, 2003, pp. 1470–1477.

[19] E. Garcia-Fidalgo and A. Ortiz, "On the Use of Binary Feature Descriptors for Loop Closure Detection," in *ETFA*, 2014, pp. 1–8.

[20] D. Gálvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Trans. Rob.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[21] M. Muja and D. G. Lowe, "Fast Matching of Binary Features," in *Conference on Computer and Robot Vision*, 2012, pp. 404–410.

[22] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," *J. Doc.*, vol. 28, pp. 11–21, 1972.

[23] P. H. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Comput. Vision Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[24] R. Mur-Artal and J. D. Tardos, "ORB-SLAM: Tracking and Mapping Recognizable Features," in *RSS*, 2014.

[25] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *ECCV*, no. 1, 2006, pp. 430–443.

[26] P. J. Burt and E. H. Adelson, "A Multiresolution Spline with Application to Image Mosaics," *ACM Trans. Graph.*, vol. 2, no. 4, pp. 217–236, 1983.

[27] F. Bonnin-Pascual, A. Ortiz, E. Garcia-Fidalgo, and J. P. Company, "A Micro-Aerial Platform for Vessel Visual Inspection based on Supervised Autonomy," in *IROS*, 2015.