



**Universitat de les Illes Balears**

Departament de Ciències  
Matemàtiques i Informàtica

---

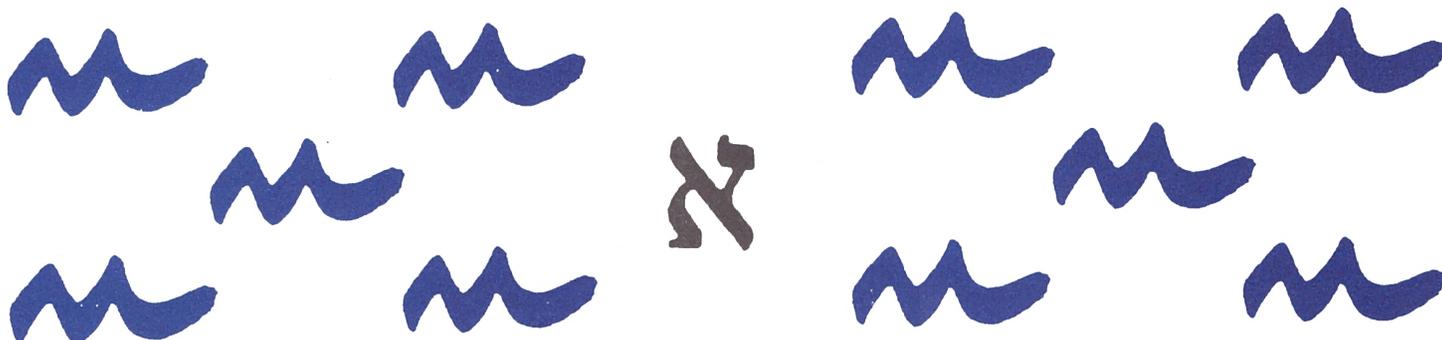
**Vessel Visual Inspection: A Mosaicing Approach**

---

---

EMILIO GARCIA-FIDALGO, ALBERTO ORTIZ,  
FRANCISCO BONNIN-PASCUAL and JOAN P. COMPANY

---



# Vessel Visual Inspection: A Mosaicing Approach

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

**Abstract**—Vessel maintenance entails periodic visual inspections of internal and external parts of the hull in order to detect the typical defective situations affecting metallic structures. Nowadays, robots are becoming more and more important regarding these inspection tasks, since they can collect the requested information and, thus, prevent humans from performing tedious, and even dangerous tasks because of places hard to reach for humans. A Micro Aerial Vehicle (MAV) fitted with vision cameras can be used as part of an automated or semi-automated inspection strategy. The resulting collection of individual images, however, does not permit the surveyor to get a global overview of the state of the surface under inspection, apart from the fact that typically defects appear broken along a number of consecutive images. Image mosaicing can certainly help in this case. To this end, in this paper, we propose a novel image mosaicing approach able to deal with this kind of scenarios. Our solution employs a graph-based registration method from which relevant topological relationships between (overlapping) images are found. This graph is built according to a visual index based on a Bag-of-Words (BoW) scheme making use of binary descriptors for speeding up the image description process. At the end of the paper, we report about the results of a number of experiments that validate our approach, including the outcome of defect detectors working directly over the mosaic.

## I. INTRODUCTION

Vessels represent one of the most common ways of transport around the world. Despite these ships do not normally suffer important maritime accidents, sometimes they occur with catastrophic consequences. Since the structural failure is the major cause of shipwrecks, vessels need to be accurately inspected to ensure their correct structural integrity. Due to the size of these vessels, the inspection process becomes a tedious, long task, which nowadays is carried out by human surveyors which in turn must be made in the shortest time possible to reduce costs. Besides, the vessel has to be emptied and situated in a dockyard, where typically temporary staging, lifts and movable platforms need to be installed to allow the workers for close-up inspection of the different metallic surfaces. In addition to the significant heights, the operational environment can include flammable and toxic gases. Under these conditions, the survey can become a very hazardous task for human operation. Furthermore, the total cost of a

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 project INCASS. This project 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

full vessel inspection, which can exceed \$1M, is directly related to the time along which the ship is inoperable.

Robots can be useful for automating the aforementioned operations, simplifying the inspection task and making it possible to collect information from places which are hard to reach for humans. The EU-funded FP7 MINOAS project [1] developed a fleet of robots to this end. Within this context, a Micro Aerial Vehicle (MAV) [2] fitted with a flexible set of cameras was proposed to provide the surveyor with an overview of the surfaces present in cargo holds. The collected images were next used as input to several algorithms capable of detecting different kinds of defects that can affect metallic surfaces, such as cracks or corrosion. During the several field tests that took place as part of the MINOAS project activities, surveyors asked for a better presentation of the visual data collected. This has been assumed as a requirement for the INCASS project (follow-up of MINOAS) in the form of a tool able to provide a global overview of the surfaces under inspection, so that defects do not appear broken along several consecutive images and the surveyor can easily get an overall impression of them.

To this end, in this paper, we propose a MAV-based novel image mosaicing approach to create seamless composite images of the area under inspection. Image mosaicing drawn the attention of the robotics community some years ago, specially for mapping areas using down-looking cameras, for e.g. underwater [3]–[8] and aerial/satellite applications [9]–[12]. However, it is less usual to find solutions that make use of forward-looking cameras [13], as it is our approach, which in turn captures the images from a MAV operating at close distance from the scene (less than 2 meters, since the surveyor needs to be within arm's reach from the hull).

More precisely, our solution employs a graph-based registration method to find relevant topological relationships between (overlapping) images. Additionally, the graph representation allows us to search for the shortest path between every image and a chosen reference image (i.e. the *mosaic frame*), minimizing the number of transformation compositions required to compute the alignment of images. To find image overlapping candidates, we employ a binary visual dictionary [14], which is based on a Bag-of-Words (BoW) scheme that is built in an online manner. Unlike most image mosaicing solutions, which make use of SIFT [15] or SURF [16] to describe images, our approach takes advantage of the use of the FAST corner detector [17] combined with the LDB binary descriptor [18] to speed up the description process. In order to illustrate the usefulness of our approach during vessel inspections, we have provided a defect detector [19] with the outcome of the mosaicing process. The



Fig. 1. An example of a cargo hold of a container ship.

resulting performance is discussed at the end of the paper

The rest of the paper is organized as follows: Section II describes the conditions under which the images are captured, Sections III and IV describe our mosaicing approach, Section V reports the experimental results obtained, and Section VI concludes the paper.

## II. OPERATING CONDITIONS

This section describes the conditions under which the images for the mosaicing are captured during a typical inspection task, as well as the typical image content, since they both define the complexity of the image mosaicing process. In our case, we consider *container ships*, i.e. vessels that carry all of their load in truck-size intermodal containers stacked in cargo holds. The vertical structures that can be found in these holds are of prime importance. To make proper repair/non-repair decisions, the surveyor must be provided, among other kinds of input, with imagery detailed enough so as to enable the remote visual assessment of these structures. During the inspection process, the platform sweeps the relevant metallic surfaces and grabs pictures at a rate compatible with its speed. The areas suspected of being defective can be re-visited for acquiring close-up images, taking thickness measurements (by means of other platforms), or even be compared in a posterior inspection. Our goal is to create a mosaic using as input the images obtained during this kind of visual inspection task.

Figure 1 shows an example of cargo hold. As can be seen, the walls look globally as more or less planar surfaces, which in this particular case measure in height around 15 meters (they can reach up to 20-25 meters). In order to obtain useful images for visual inspection, the vehicle needs to fly at short distance from the walls (less than 2 meters). Because of both the fast dynamics of the MAV and the forward-looking camera configuration, this fact does not lead to favourable image capture conditions, contrary to other mosaicing scenarios where the effects of platform oscillations are attenuated by the distance between camera and scene, or simply they are negligible.

## III. MOTION MODEL

The motion model plays a key role in the image registration process. In this work, the camera is assumed to be perpendicular to the scene, which, as said previously, can be considered more or less planar. Under these conditions, two overlapping images  $I_i$  and  $I_j$  are related by a homography, a linear transformation represented by a  $3 \times 3$  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 our approach can deal with affine transformations (six degrees of freedom), the motion of the vehicle can be approximated by a simpler model using similarity transformations, which has four degrees of freedom comprising rotation, translation and scaling.  ${}^iH_j$  is expressed as:

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

where  $s$  is the scale,  $\theta$  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 (MLE-SAC) [20] is next used as a robust estimation algorithm to minimize the reprojection error for (1) and discard outliers. Finally, given a path  $I_i, I_{k_1}, \dots, I_{k_m}, I_j$ , the associated transformation that relates frames  $I_i$  and  $I_j$  is computed by concatenating the corresponding relative homographies  ${}^iH_j = {}^iH_{k_1} {}^{k_1}H_{k_2} \dots {}^{k_{m-1}}H_{k_m} {}^{k_m}H_j$ .

## IV. THE MOSAIC BUILDING PROCESS

A mosaic is created by aligning images according to a selected frame, commonly referred to as the *mosaic frame*, and blending them together in a larger composite. This comprises a sequence of more elementary steps, which, for our approach, is outlined in Fig. 2. As usual, the process starts with the description of the images, which in our case is performed using binary descriptors for efficiency reasons. The next step discards those images which are not going to contribute significantly to the mosaic by adopting a keyframe-based approach. Subsequently, the surviving images are indexed by means of the computed binary descriptors; this permits, on the basis of image similarity metrics, to detect relevant overlap between images [14]. This allows us to estimate the topological relationships between the different images, from which we apply bundle adjustment to optimize the image alignment. Finally, in accordance to the resulting alignment, the last step produces the mosaic by blending together the image pixels contributing to the same mosaic pixels.

We use thus a *global alignment* method in order to avoid the drift introduced by concatenating homographies estimated from consecutive frames, although at some steps we perform *pairwise alignment*. In our case, to implement this global alignment process, we use exclusively visual information, from which we infer image similarities that suggest significant overlap between images, either consecutive

or non-consecutive in the image sequence. Since a frame-to-frame comparison approach easily becomes unfeasible as the number of input images grows, we index every image using binary descriptors. This index is used next to determine the similarity between images and establish topological relationships between them. From this, we build a graph structure which allows us to find the shortest paths between the mosaic frame and any other image [7], [21], [22], in order to reduce the number of transformation compositions and further improve the alignment of every frame with the mosaic. Bundle adjustment follows next on the basis of the data finally available in the graph structure.

The following sections detail the aforementioned steps:

1) *Image Description*: This step computes a compact representation of every image as a collection of FAST features [17] described each by an LDB binary descriptor [18]. This allows us to take advantage of their faster computation times and reduced storage needs, in front of the classic approaches used previously in image mosaicing such as SIFT [15] and SURF [16]. Besides, to favour accurate estimations 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. To this end, a  $4 \times 4$  regular grid is defined over the image and around 190 features are expected from each cell.

In order to ensure this minimum number of features, the detector is allowed to iterate around the feature detection parameters until the requested number of features is available. To reduce the number of iterations, the contrast of the sequence is globally improved by transforming each image to HSV space, computing the V-channel histogram for the full set of images, deriving an intensity look-up table (LUT) that stretches this histogram, and finally transforming each V-level at each image according to the computed LUT, together with going back to the RGB colour space.

2) *Image Selection*: This step discards images which are not deemed to provide a significant contribution to the mosaic in order not to introduce unnecessary drift during the alignment process. This contribution is measured as the amount of overlap between the current image and a previous image considered as a keyframe, so that the higher the overlap the less relevant is the image. More specifically, we compute the homography between both images, warp the four corners of the current image onto the keyframe, take the maximum distance between each pair of corresponding corners and discard the frame if this distance is below a certain threshold. In order to relate the magnitude of the camera motion to the size of the input images, the distance and the threshold are normalized by the maximum of the horizontal and vertical image sizes.

3) *Image Indexing*: The goal of this step is to build an image index for, given a query input, finding similar images. This index, inspired by image retrieval methods, is an efficient way to determine overlapping image pairs. Image retrieval methods developed recently are based on the BoW approach [23]. 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 [15], [16] and is less common to find binary solutions [24]. In this work, we employ a method for computing a vocabulary of binary features that can be built online, avoiding thus a training phase. A brief overview can be found next (the interested reader is referred to [14] for the details).

Our method is based on an incremental visual dictionary based on a modified version of Muja and Lowe’s approach [25]. The dictionary is combined with an inverted index, which contains, for each 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. We propose to use a bitwise AND operation. Formally, being  $B$  a binary descriptor:

$$B_{w_i}^t = B_{w_i}^{t-1} \wedge B_q, \quad (2)$$

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.

In order to save storage space and computation time, only a subset of the total features, corresponding to the image features with higher response, is used for indexing each image. In our experiments, around 500 features have been enough to find most of the overlaps. The index is initially built using the descriptors of the first image. 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 [15] in order to determine if both descriptors represent the same visual feature. If positive, the query descriptor and the visual word are merged using (2) and 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 feature. Given the features of a query image as input, the visual index returns an ordered list of images according to a scoring process based on Term Frequency Inverse Document Frequency (TF-IDF) weighting [26].

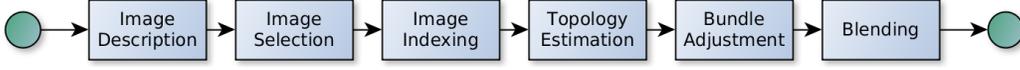


Fig. 2. Steps performed in our image mosaicing approach.

4) *Topology Estimation*: The topology of the environment represents the relationships that exists between the images of the surveyed area. This topology is modelled by means of an undirected graph, whose nodes represent the individual images and edges represent overlaps between them. As a measure of the quality of a link, an edge between the images  $I_i$  and  $I_j$  is labelled with the following weight  $w$ :

$${}^i w_j = \frac{\sum_{k=1}^n \|p_i^k - {}^i H_j p_j^k\|}{n}, \quad (3)$$

where  $n$  is the number of inliers obtained during the computation of  ${}^i H_j$  and  $(p_i^k, p_j^k)$  are the corresponding points for the inlier  $k$ .

The construction of the graph relies on the index built in the previous step, running iteratively for each image. First, a link is added between the current image and the next one. Then, the image is queried against the index for obtaining a list of similar images ordered by their TF-IDF value, discarding the current image and the next one, since a link has already been added between them. For the top  $c$  candidates (15 in our experiments), we compute their homographies with the current frame, and rerank all the candidates in accordance to the number of inliers. If this number is higher than a certain threshold (500 in our experiments) and the images pass an overlap spatial verification step based on the intersection between the circles circumscribing the respective warped frames [22], a link between those images is incorporated into the graph.

Once the graph has been built, all the images are transformed to a common reference frame. This transformation is performed by means of an *absolute homography*  ${}^M H_i$ , which relates image  $i$  with the reference frame. The image corresponding to the node of the graph with the highest output degree is selected as the reference or *mosaic frame*, whose absolute homography is thus the identity matrix.

Finally, the shortest-path tree (SPT) rooted at the mosaic frame is found using Dijkstra's algorithm. For each of the remaining images, the absolute transformation is computed by traversing the graph using the shortest path from the root node to the image, and concatenating the corresponding pairwise homographies. This allows us to find the absolute homographies using the minimum number of transformation compositions for each case.

5) *Bundle Adjustment*: Despite the efforts for accurately estimating the images relationships, alignment errors still arise, resulting into a globally inconsistent map. To correct this problem, a bundle adjustment step is performed in order to jointly minimize the global misalignment induced by the current absolute homographies. The error function is defined

as follows:

$$\epsilon = \sum_i \sum_j \sum_{k=1}^n \|p_i^k - ({}^M H_i)^{-1} {}^M H_j p_j^k\| + R({}^M H_j) \\ \|p_j^k - ({}^M H_j)^{-1} {}^M H_i p_i^k\| + R({}^M H_i), \quad (4)$$

where  $i$  and  $j$  are two images related by a link which belongs to the SPT,  $n$  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 we assume that the vehicle flies at a more or less constant distance from the wall, and are defined as follows:

$$R({}^M H_i) = \gamma (a^2 + b^2 - 1) = \gamma ((s \cos \theta)^2 + (s \sin \theta)^2 - 1) \quad (5)$$

where  $\gamma$  is a regularization factor,  $s$  and  $\theta$  are the, respectively, scale and orientation contained in the homography, and  $a$  and  $b$  are defined in (1). To reduce the influence of outliers, we optimize, instead of (4), 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<sup>1</sup> and the absolute homographies available so far as a starting point. Usually a few iterations are needed to achieve convergence, resulting into better estimations of the absolutes homographies.

6) *Blending*: As a last step, the final mosaic is created using the multi-band blending algorithm [27] to diminish the visual artifacts that result from the combination of the images contributing to the mosaic. This step makes use of the *stitching* module implemented in the OpenCV library. Besides the multi-band blending, this module also includes seam finding and exposure compensation, which perfectly suits our needs.

## V. EXPERIMENTAL RESULTS

We have validated our approach under different operating conditions. The experiments are summarized in Table I, indicating, for each case, the total number of images in the input set, the total number of images selected by our algorithm, the initial cost of the error function, the final cost after the bundle adjustment step, the number of iterations needed to achieve convergence, the time needed to create the mosaic and the average reprojection error measured as the ratio between the final cost and the resulting non-zero pixels of the mosaic. As shown, the optimization step

<sup>1</sup><http://ceres-solver.org/>

TABLE I  
SUMMARY OF EXPERIMENTAL RESULTS

Seq	#Imgs	Sel	IC	FC	I	T	Avg
SYNTH1	679	331	$8.78e^6$	$2.43e^5$	51	649.4	0.024
SYNTH2	431	86	$1.77e^6$	$3.93e^4$	165	68.0	0.036
FLIGHT1	137	54	$3.04e^5$	$1.17e^4$	87	108.6	0.014
FLIGHT2	274	25	$1.35e^6$	$7.14e^4$	56	94.0	0.027
FLIGHT3	143	64	$1.84e^5$	$4.90e^4$	84	152.4	0.061
FLIGHT4	517	53	$8.33e^5$	$3.11e^4$	75	48.0	0.011
UNW	201	86	$7.43e^5$	$2.88e^4$	116	187.5	0.092
ODEMAR	65	47	$1.64e^6$	$9.77e^4$	66	42.2	0.095
BBOARD	291	14	$2.47e^5$	$2.44e^4$	75	45.0	0.047

*#Imgs* : Number of images in the input set.  
*Sel* : Number of images selected by our approach.  
*IC* : Initial cost before the bundle adjustment step.  
*FC* : Final cost after the bundle adjustment step.  
*I* : Number of iterations needed to converge.  
*T* : Time in seconds.  
*Avg* : Average reprojection error in pixels.

reduces the global misalignment of the images, improving the registration of the images prior to blending the final mosaic. Notice that our approach is able to achieve good results despite it discards a high number of frames.

For the experiments, we used the  $2.5 \times 4$  meter canvas that can be seen in Fig. 3, which directly comes from a real wall of a cargo hold of a container ship. Further, to get more insight on the performance of the mosaicing approach, we have tested other sequences recorded from other environments, e.g. underwater, whose results are also discussed.

In a first kind of experiment, we generated synthetic sweeping trajectories over a full image of the canvas, introducing random alterations in scale, rotation and translation, collecting subimages of size  $640 \times 480$  from time to time. The synthetic sequence that Table I refers to as *SYNTH1* comprised a total set of 679 images which were introduced as input to the mosaicing algorithm. A subset of 331 images were selected from the original set. The resulting mosaic can be found in Fig. 4, while the estimated topology is shown in Fig 5. As can be seen, the alignment in this case is correct, generating the original image in a seamless mosaic despite the the simulated MAV motion. An additional squared trajectory introducing random motion was generated over an underwater image (referred as *SYNTH2*), producing 431 images of size  $320 \times 240$ . The resulting mosaic and the estimated topology are shown, respectively, in Fig. 6 and Fig. 7. Note that the shape in the corners are because our approach estimates the motion using the top-left corner of an image as the origin of a coordinate system.

For a second kind of experiments, we flew in front of the canvas a real MAV based on the AscTec Hummingbird platform, fitted with a  $752 \times 480$ -pixel/ $58^\circ$ -lens uEye UI-1221LE camera running at 10Hz. Notice that this is a more challenging situation because of the motion of the MAV at close distance to the canvas, resulting into a more difficult image registration problem. We performed four flights doing



Fig. 3. Canvas used in our experiments for simulating a cargo hold wall.



Fig. 4. Mosaic resulting from the synthetic sweeping trajectory generated over the canvas image.

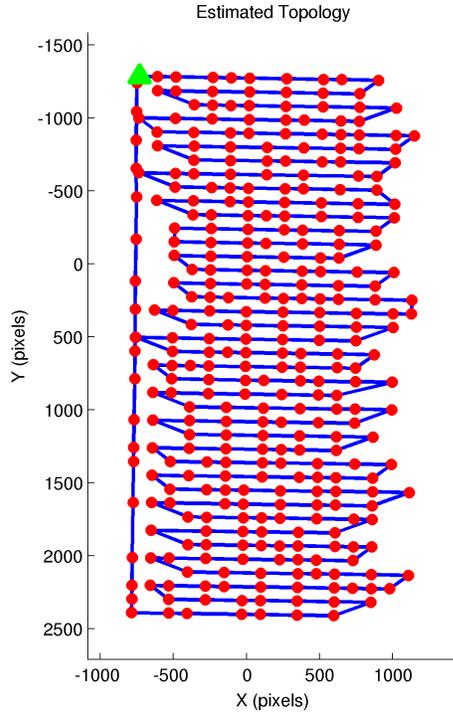


Fig. 5. Topology estimated by our approach for the synthetic sequence. Each image is indicated using a red circle and the first image is labelled by a green triangle.

different shapes, referred as *FLIGHT1* (top-down), *FLIGHT2* (square), *FLIGHT3* (free) and *FLIGHT4* (eight-shaped). All of them were recorded at a distance of 1-1.5 m. The final mosaics are shown from Fig. 8 to Fig. 11. As before, good alignment results are observed, and the defects (corrosion in this case), do not appear broken, as expected. Indeed, in order to check whether the mosaics were useful for defect detection, we supplied the composite images to the defect detector described in [19], which combines contrast and symmetry information within a probabilistic framework for corrosion detection. The results are shown from Fig. 8 to Fig. 11, where the defects are labelled in black. As can be observed, the main defective areas are correctly detected. The estimated topologies for each case are shown from Fig. 12 to Fig. 15.

In order to further validate our mosaicing approach, in a third kind of experiments, we provided the algorithm with sequences from other environments. For the particular case of the sequence referenced in Table I as *UNW*, the images come from Valldemossa harbour seabed (Mallorca, Spain) and a hand-held down-looking camera. The dataset consists of 201 images of  $320 \times 180$  pixels, which comprises a large loop, what allows us to validate the ability of our algorithm for recognizing previously seen places. A total of 86 images were selected by the algorithm, leading to the final mosaic and the estimated topology shown in Fig. 16 and in Fig. 17. The alignment, in this case, is even better than for the aerial sequences, since the camera motion is

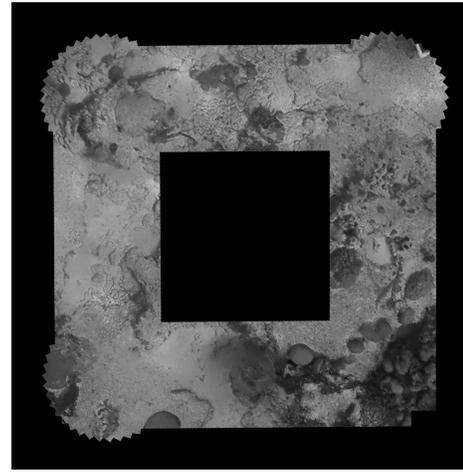


Fig. 6. Mosaic resulting from the synthetic squared trajectory generated over the underwater image.

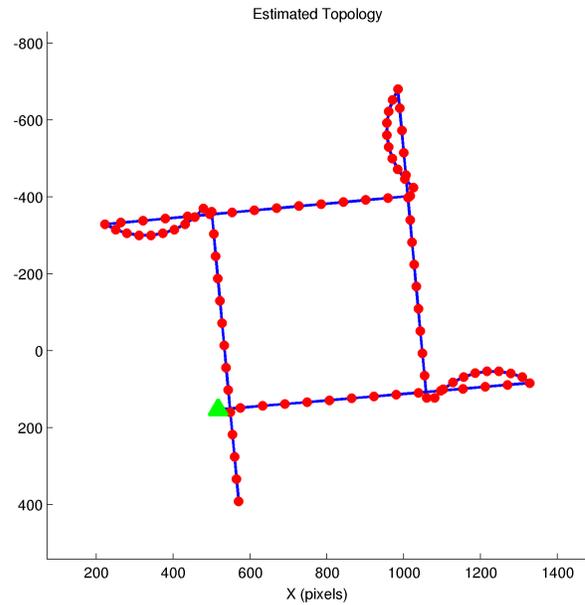


Fig. 7. Topology estimated by our approach for the squared sequence using the underwater image. Each image is indicated using a red circle and the first image is labelled by a green triangle.

less aggressive and the image registration process is easier. Results for another underwater sequence using a forward-looking camera (referred as *ODEMAR*) are also presented in Fig. 18 and Fig 19, and for a sequence including a blackboard using a hand-held camera (referred as *BBOARD*), in Fig. and Fig.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have described a novel approach for generating mosaics from images collected by a MAV, to be used for vessel visual inspection. Our scheme makes use of a graph for modeling image relationships, which allows us to determine the shortest path between each image and the mosaic frame. This graph is created using a visual



Fig. 8. (left) Mosaic using the images collected by the MAV in a top-down trajectory [mosaic contrast has been tuned for visualization purposes]. (right) Defective areas found, shown in black.

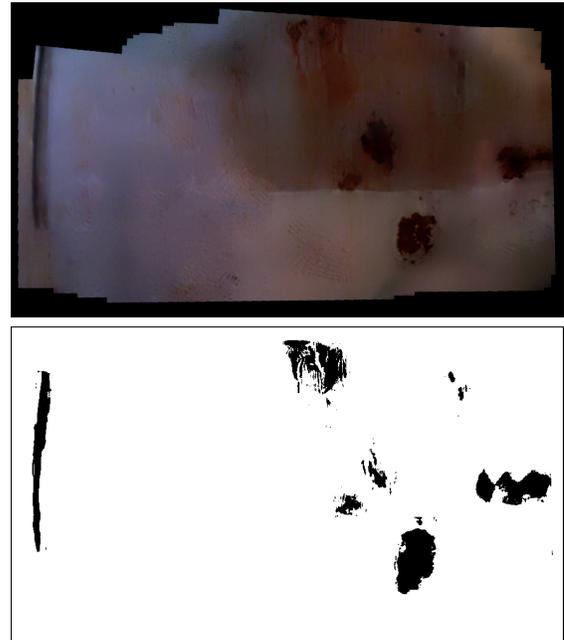


Fig. 10. (top) Mosaic using the images collected by the MAV in a free trajectory [mosaic contrast has been tuned for visualization purposes]. (bottom) Defective areas found, shown in black.

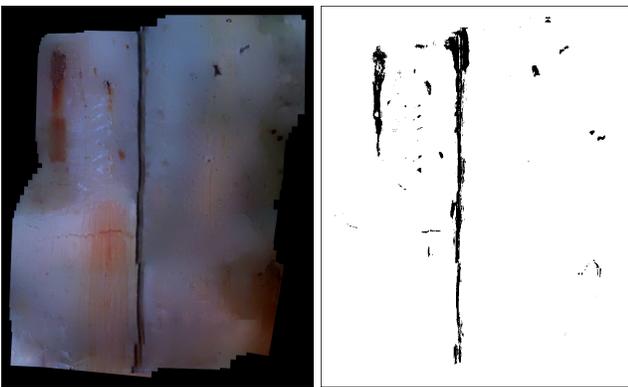


Fig. 9. (left) Mosaic using the images collected by the MAV in a squared trajectory [mosaic contrast has been tuned for visualization purposes]. (right) Defective areas found, shown in black.

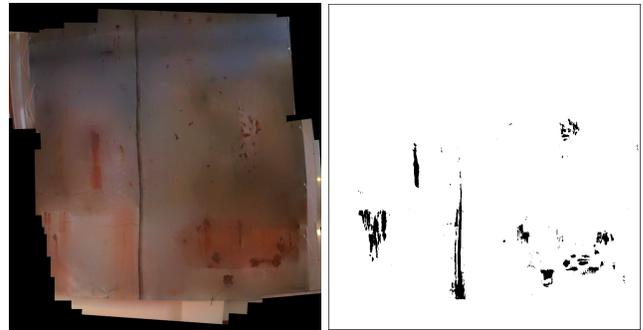


Fig. 11. (left) Mosaic using the images collected by the MAV in an eight-shaped trajectory [mosaic contrast has been tuned for visualization purposes]. (right) Defective areas found, shown in black.

index of binary features which is built online and is used to obtain, in an efficient way, overlapping candidates according to the image information. We have validated our approach under different operating conditions, concluding that the resulting mosaic can be employed to find defective situations susceptible to appear in a vessel. Despite the unfavourable MAV operating conditions during a vessel visual inspection, our approach works reasonably well, being helpful as part of an assistance tool suite for surveyors.

As part of the activities of the INCASS project, our future plans are to finish the validation of our approach onboard a real vessel. Other kinds of vessels will also be considered. Finally, we are interested in improving the image registration process by adopting a local submosaic-based strategy.

## 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," vol. 31, no. 2, pp. 319–341, 2014.
- [2] F. Bonnin-Pascual, E. Garcia-Fidalgo, and A. Ortiz, "Semi-Autonomous Visual Inspection of Vessels Assisted by an Unmanned Micro Aerial Vehicle," 2012, pp. 3955–3961.
- [3] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Mosaic-Based Navigation for Autonomous Underwater Vehicles," vol. 28, no. 4, pp. 609–624, 2003.
- [4] O. Pizarro and H. Singh, "Toward Large-Area Mosaicing for Underwater Scientific Applications," vol. 28, no. 4, pp. 651–672, 2003.
- [5] H. Madjidi and S. Negahdaripour, "Global Alignment of Sensor Positions with Noisy Motion Measurements," vol. 21, no. 6, pp. 1092–1104, 2005.
- [6] A. Elibol, R. Garcia, and N. Gracias, "A New Global Alignment Approach for Underwater Optical Mapping," *Ocean Engineering*, vol. 38, no. 10, pp. 1207–1219, 2011.
- [7] A. Elibol, N. Gracias, R. Garcia, A. Gleason, B. Gintert, D. Lirman, and P. Reid, "Efficient Autonomous Image Mosaicing with Applications to Coral Reef Monitoring," *Workshop on Robotics for Environmental Monitoring (IROS)*, 2011.

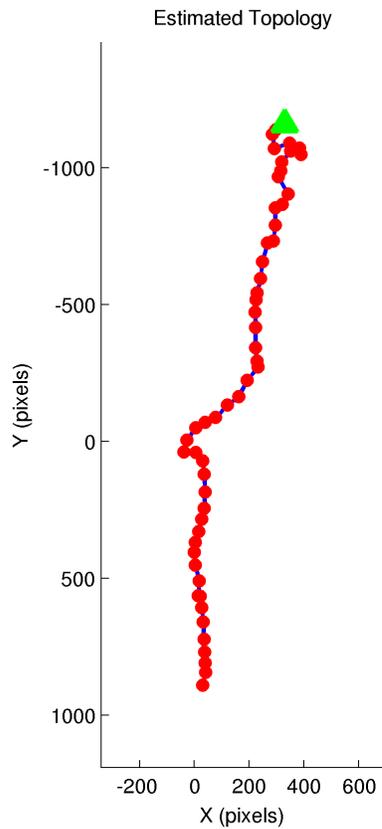


Fig. 12. Topology estimated by our approach for the flight performed by the MAV in a top-down trajectory. Each image is indicated using a red circle, and the initial image is labelled using a green triangle.

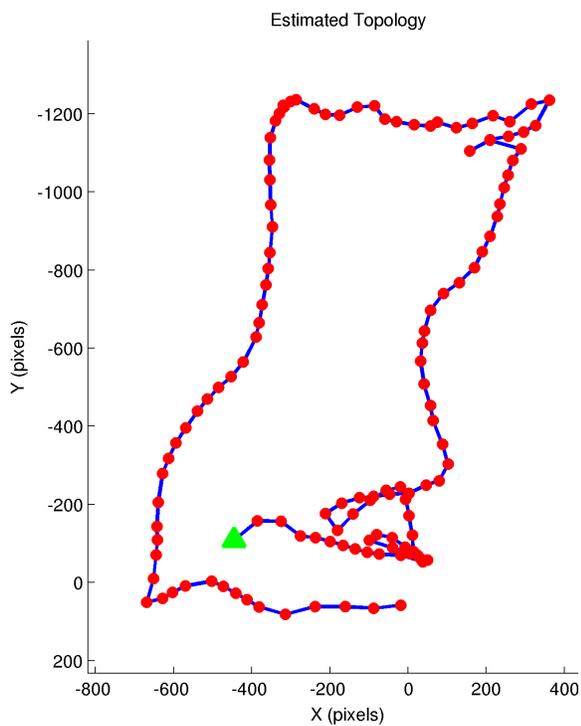


Fig. 13. Topology estimated by our approach for the flight performed by the MAV in a squared trajectory. Each image is indicated using a red circle, and the initial image is labelled using a green triangle.

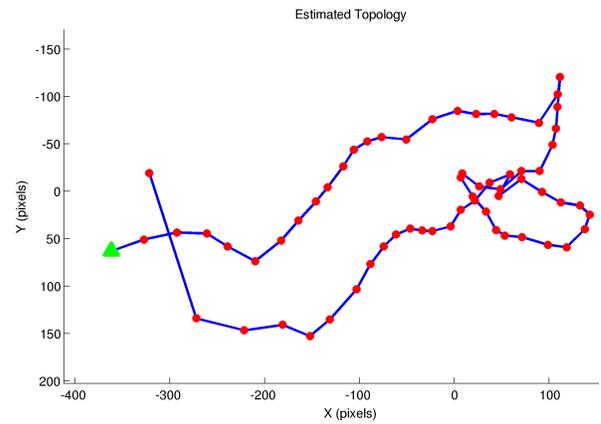


Fig. 14. Topology estimated by our approach for the flight performed by the MAV in a free trajectory. Each image is indicated using a red circle, and the initial image is labelled using a green triangle.

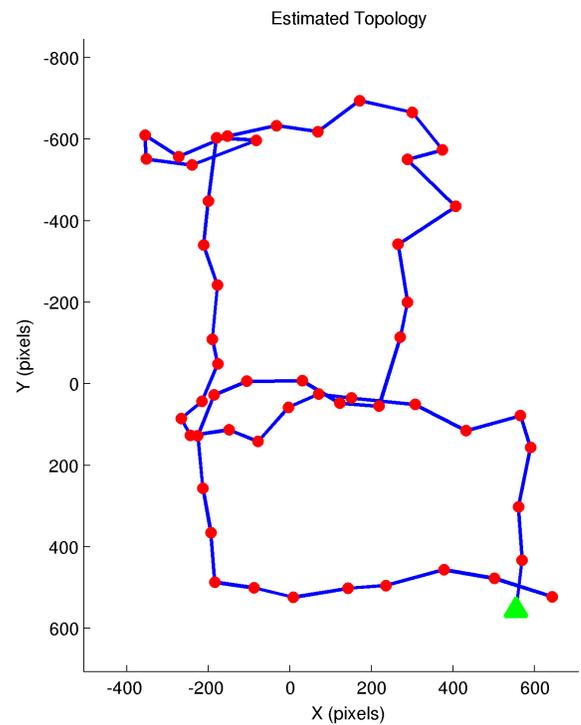


Fig. 15. Topology estimated by our approach for the flight performed by the MAV in a eight-shaped trajectory. Each image is indicated using a red circle, and the initial image is labelled using a green triangle.

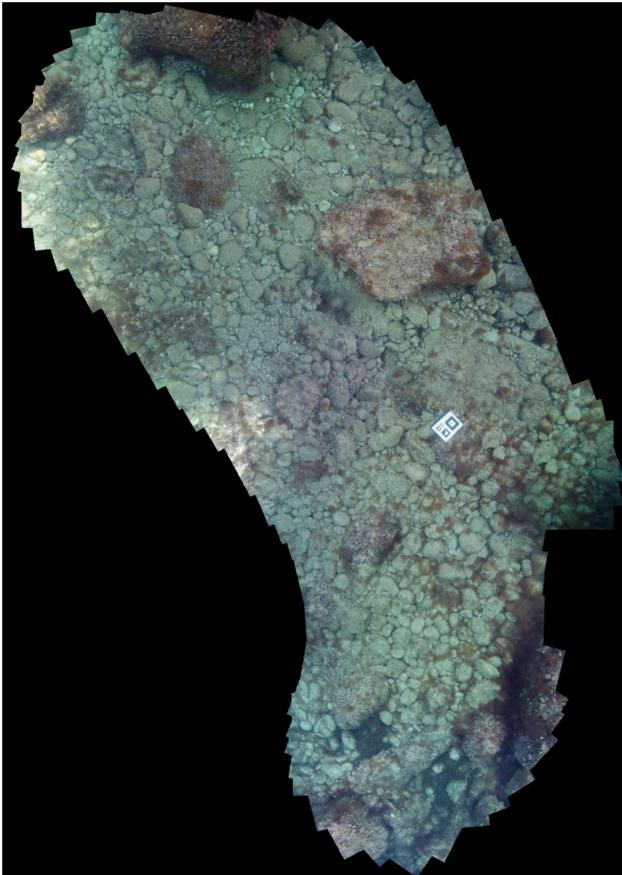


Fig. 16. Resulting mosaic obtained for the underwater dataset.

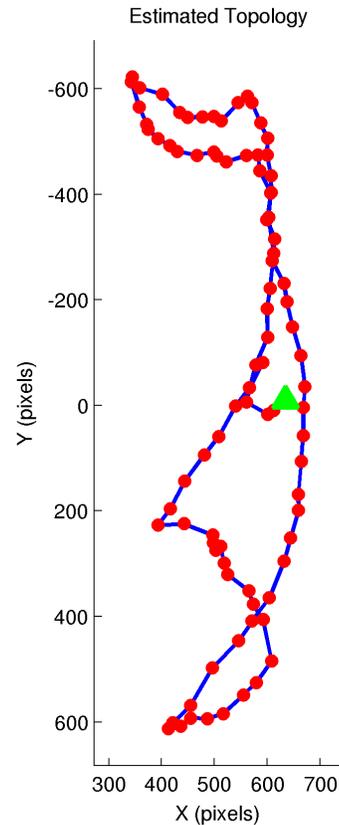


Fig. 17. Topology estimated by our approach for the underwater dataset. Each image is indicated using a red circle, and the initial image is labelled by a green triangle.

- [8] F. Ferreira, G. Veruggio, M. Caccia, E. Zereik, and G. Bruzzone, "A Real-Time Mosaicking Algorithm Using Binary Features for ROVs," 2013, pp. 1267–1273.
- [9] Y. Lin and G. Medioni, "Map-Enhanced UAV Image Sequence Registration and Synchronization of Multiple Image Sequences," 2007, pp. 1–7.
- [10] T. Kekec, A. Yildirim, and M. Unel, "A New Approach to Real-Time Mosaicking of Aerial Images," vol. 62, no. 12, pp. 1755–1767, 2014.
- [11] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Homography Based Kalman Filter for Mosaic Building. Applications to UAV Position Estimation," 2007, pp. 2004–2009.
- [12] H. Bulow and A. Birk, "Fast and Robust Photomapping with an Unmanned Aerial Vehicle (UAV)," 2009, pp. 3368–3373.
- [13] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, "Visual Inspection of Hydroelectric Dams using an Autonomous Underwater Vehicle," vol. 27, no. 6, pp. 759–778, 2010.
- [14] E. Garcia-Fidalgo and A. Ortiz, "On the Use of Binary Feature Descriptors for Loop Closure Detection," 2014, pp. 1–8.
- [15] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," vol. 60, no. 2, pp. 91–110, 2004.
- [16] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," vol. 3951, 2006, pp. 404–417.
- [17] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," no. 1, 2006, pp. 430–443.
- [18] X. Yang and K.-T. Cheng, "Local Difference Binary for Ultrafast and Distinctive Feature Description," vol. 36, no. 1, pp. 188–94, 2014.
- [19] F. Bonnín-Pascual and A. Ortiz, "A Probabilistic Approach for Defect Detection based on Saliency Mechanisms," 2014, pp. 1–4.
- [20] P. H. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," vol. 78, no. 1, pp. 138–156, 2000.
- [21] H. S. Sawhney, S. Hsu, and R. Kumar, "Robust Video Mosaicking through Topology Inference and Local to Global Alignment," 1998, pp. 103–119.
- [22] R. Marzotto, A. Fusiello, and V. Murino, "High Resolution Video Mosaicking with Global Alignment," vol. 1, 2004, pp. I-692–I-698 Vol.1.
- [23] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," 2003, pp. 1470–1477.
- [24] D. Gálvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," vol. 28, no. 5, pp. 1188–1197, 2012.
- [25] M. Muja and D. G. Lowe, "Fast Matching of Binary Features," in *Conference on Computer and Robot Vision*, 2012, pp. 404–410.
- [26] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.
- [27] 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.

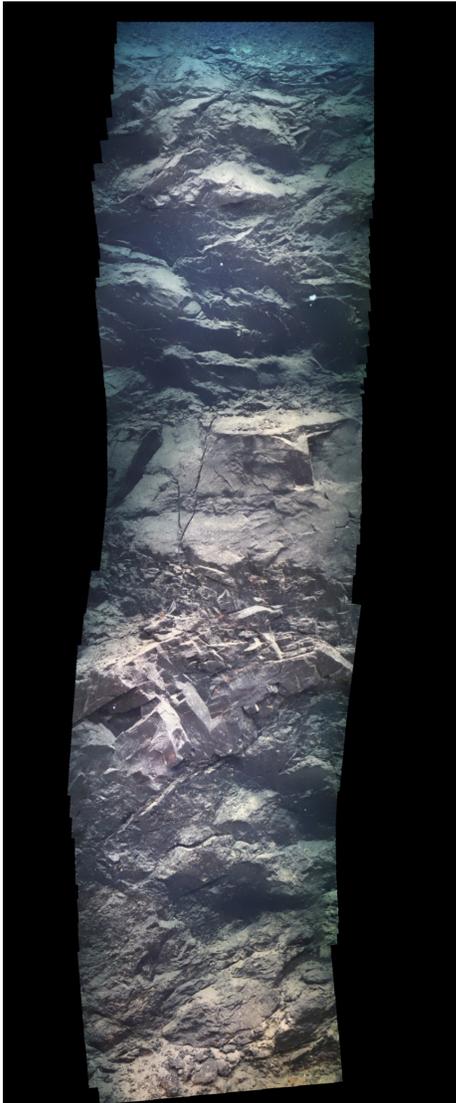


Fig. 18. Mosaic resulting from the ODEMAR sequence.

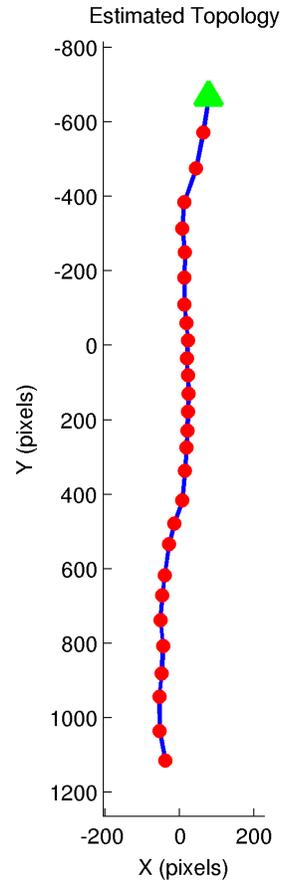


Fig. 19. Topology estimated by our approach for the ODEMAR dataset. Each image is indicated using a red circle and the first image is labelled by a green triangle.

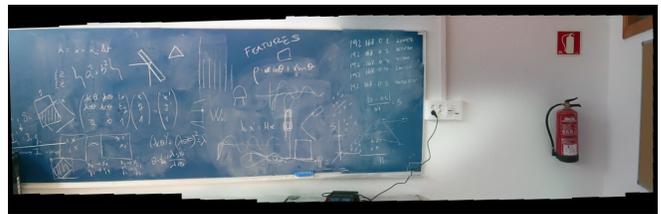


Fig. 20. Mosaic resulting from the BBOARD sequence.

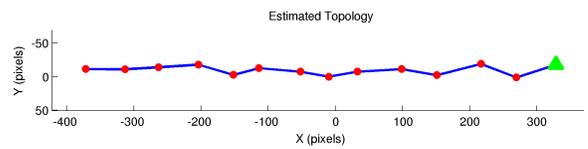


Fig. 21. Topology estimated by our approach for the BBOARD dataset. Each image is indicated using a red circle and the first image is labelled by a green triangle.