

# LAPMv2: An Improved Tool for Underwater Large-Area Photo-Mosaicking

Yann Marcon

HGF-MPG Research Group on Deep-Sea Ecology and Technology  
Alfred-Wegener-Institut Helmholtz-Zentrum für Polar- und Meeresforschung, 27570 Bremerhaven, Germany  
E-mail: Yann.Marcon@awi.de

**Abstract**—This paper presents a research software solution specifically developed to allow marine scientists to produce geo-referenced visual maps of the seafloor, known as mosaics, from a set of underwater images and navigation data. LAPMv2 is a suite of tools, which provides the users with the current state-of-the-art methods for automatic feature detection and matching, as well as with powerful tools for registering images, constructing and geo-referencing photomosaics, and importing them into geographic information systems such as ArcGIS. The main key strengths of LAPMv2 include (1) its robust false-match rejection method, which allows the application to run fully autonomously without producing aberrant results caused by erroneous matches, (2) the total control of the user over the mosaicking workflow, (3) the possibility to intuitively create matches between unlinked images, and (4) the graphical interface that guides the user through the different steps of the mosaicking workflow.

**Keywords**—*Photo-mosaicking; photomosaic; underwater imagery; seafloor mapping; GIS*

## I. INTRODUCTION

Due to the poor visibility in the deep sea ( $> 200\text{m}$ ), and to the strong absorption of light underwater, images of the seafloor have to be taken from a very low altitude above the sea bottom ( $< 10\text{m}$ ). Such short distance strongly limits the field of view of the camera and, in order to image a larger area of the seafloor, several adjacent and overlapping images need to be taken. The process of mosaicking consists of aligning and combining those images together in order to generate a large comprehensive image of the mapped area.

LAPMv2 is a research software solution dedicated to the routine production of underwater photomosaics. It was developed with the aim of making the process of photo-mosaicking accessible to every end-users without requiring in-depth knowledge of the technical aspects. To that end, a graphical user interface (GUI) was created in order to make the mosaicking workflow intuitive for the users. However, the ease of use was not traded off for efficiency, and LAPMv2 is based on well proven methods for feature detection and matching [1], [2], as well as on robust bundle adjustment techniques [3].

Several works have been published in the field of 2D mosaicking, and impressive results have been achieved [4]–[7], [3], [8]–[11]. The LAPMv2 is not reinventing the wheel, and is strongly inspired by some of that existing work [3], [9]. Although there are some variations, which are presented in this paper, the main motivation for developing the LAPMv2 was to provide marine scientists with an end-user tool that would

allow them to intuitively and routinely create large and geo-referenced photomosaics. LAPMv2 is meant to be freely available to user scientists.

Despite a similar-looking user interface, LAPMv2 has very little to do with its predecessor [11]. The core structure of the tool was re-written entirely in order to allow for more flexibility and consistency in the code, and to enable the implementation of new functionalities such as 3D camera pose estimation, use of fiducial points [9] and fiducial pairs, and an improved global registration method. However, one of the main strengths of LAPMv2 resides in its ability to run through the entire photo-mosaicking workflow entirely autonomously. Such ability is enabled by both an automatized search for overlap between non-consecutive images (sidelap), which guarantees a better accuracy in the image alignment, and an extremely robust false-match rejection method, which ensures that all erroneous matches are filtered out, thus preventing aberrant results in the final mosaics. Nevertheless, if required, users also have the possibility to supervise and fine-tune any step of the mosaicking workflow, in order to further improve the results. Furthermore, in terms of input data, the tool is compatible with most current underwater imaging systems. Besides the image dataset, LAPMv2 requires some additional data such as the navigation data of the camera system, the time code of the images and the camera intrinsic and extrinsic parameters. Those data are nowadays available for most underwater photo- and video-imaging systems.

This work presents the LAPMv2 tool and is structured as follow: Part II introduces the user interface and the workflow of the program, Part III focuses on the inputs and outputs data, Part IV describes and explains the different algorithms that are used, and Part V discusses the results and strengths of LAPMv2. The terminology used throughout this paper agrees with the following definitions:

- *Feature*: detected SIFT feature [1], [2].
- *Match*: matched features.
- *Inlier*: valid match (after false-match rejection).
- *Outlier*: erroneous match (or false match).
- *Link* (or *matched pair*): image pair, between which valid matches have been found.
- *Sidelap*: overlap between images that are not consecutive in the image sequence [11].

## II. GRAPHICAL USER INTERFACE AND EASE OF USE

### A. Mosaicking workflow

The control of the mosaicking workflow is done entirely through a graphical user interface (Fig. 1). Indeed, the initial purpose of LAPMv2 was to provide scientists with an intuitive tool for creating large geo-referenced photomosaics of the seafloor. The motivation for this was that end-user marine scientists do not necessarily possess the technical know-how or the mathematical and programming knowledge that are required in order to apply published photo-mosaicking methods themselves. By providing an intuitive GUI, LAPMv2 aims at making the process of photo-mosaicking accessible to any end-user scientists.

To that end, the design of the GUI is intended to guide the user through the different steps of the workflow. The mosaicking workflow of LAPMv2 comprises the following steps:

1. Navigation filtering and smoothing (optional).
2. Search for overlap in the image sequence: feature detection and matching between images that are consecutive in the given order of the image sequence.
3. Search for *sidelap*, i.e. overlap between non-consecutive images within a defined radius around each image.
4. Linear global registration.
5. Iterative repetition of steps 2 and 3 until no additional sidelap is found.
6. Non-linear global registration (optional).
7. Mosaic construction.

Each step of the workflow can be run separately. Nevertheless, since the computing times of some steps can take very long, the tool also gives the possibility to run all steps in one go. In such case, the parameters for each individual step can be set up beforehand, and the entire mosaicking process is batched. The methods and algorithms used in the different steps are described in Section IV.

### B. Default values and fine-tuning

Besides the ease of use, options for controlling the different steps of the mosaicking workflow are manifold. In particular, the control options of LAPMv2 allow users to

- modify the parameters of the feature detection and outlier rejection,
- filter and smooth the navigation data (Fig. 2),
- visualize and edit matches between any pair of images,
- manually create matches between any pair of images (particularly useful for cases where the images do overlap but the automatic matching fails),
- define the search radius for sidelap detection,

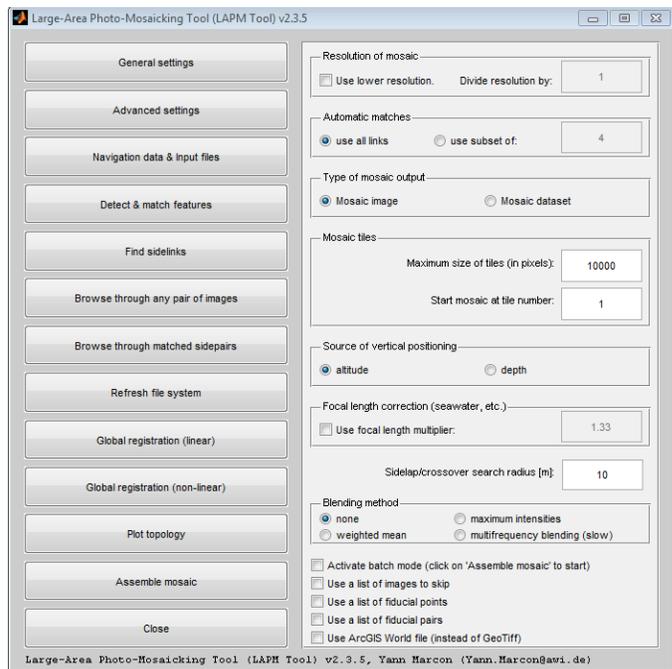


Fig. 1. Graphical user interface of LAPMv2. From the top to the bottom, the button sequence on the left panel guides the user through the steps of the mosaicking workflow.

- adjust the weighting coefficients of the cost function used in the global registration,
- review the results of the different registration methods before starting the computation of the mosaic,
- choose between different rendering methods of the final mosaic (no blending, maximum intensities, weighted mean, multi-frequency-band blending), and
- build the mosaic in different resolutions.

Default values for all customizable parameters were chosen experimentally after having tested the LAPMv2 tool on several datasets of various sizes. The selected values generally work well with most underwater datasets. However, each dataset is unique and minor tweaks in the default settings may in some cases be necessary to refine the mosaicking process. For instance, the default number of levels used by LAPMv2 for the SIFT detection is set to a higher value (6 instead of 3) than the default value suggested in [12], in order to improve the robustness of the feature detection and matching. The reason behind this is that images of the deep (>200 m) seafloor have lower contrasts and fewer features than the type of imagery data used in [12]. Similarly, the default values for the outlier rejection algorithm have been selected after extensive testing. The detailed function of each individual parameter is explained in the user guide of LAPMv2. Briefly put, the chosen settings make the false-match rejection very strict in order to allow for LAPMv2 to run autonomously. However, by lowering the strictness of the settings, the total amount of detected matched

pairs may be increased. Finally, the parameters of the global registration are explained in Section IV.

### III. INPUTS AND OUTPUTS

#### A. Inputs of LAPMv2

Whilst the term *video-mosaicking* refers to mosaicking techniques that run on video fluxes, the term *photo-mosaicking* is not restricted to mosaics made of still *photographs*, but more generally to mosaics made of still *images*. This means that photo-mosaicking techniques can also be applied to a set of video frames. One advantage of working with video frames is that the amount of overlap between consecutive images can be controlled by setting the time interval, at which frames are extracted from the video. For simplicity and compatibility reasons, LAPMv2 does not accept video input directly. Instead, the frame sequence should be extracted from the video first and then be provided to the program as a set of still images.

In that regard, LAPMv2 can run on any imagery dataset, provided that the following additional data are also available:

- Navigation data of the camera platform (ROV, AUV, etc.), including at least the geographic coordinates, the heading, and the altitude above seafloor. If available, pitch and roll angles can also be provided.
- Date and time each image was taken (timestamps).
- Order of the image sequence (in some cases the image order in the mosaicking survey is not chronological).
- Camera intrinsic parameters (e.g. focal length, sensor dimensions).
- Camera pose parameters (decentering parameters and orientation angles).

However, those types of input data are nowadays standard and are available with most ROV- and AUV-borne underwater imaging systems. Hence, the LAPMv2 tool should be compatible with most current systems.

In terms of geo-referencing data format, LAPMv2 uses the WGS84 Geographic Coordinate System for both input and output files. Indeed, underwater systems commonly use acoustic positioning systems, such as ultra-short baseline

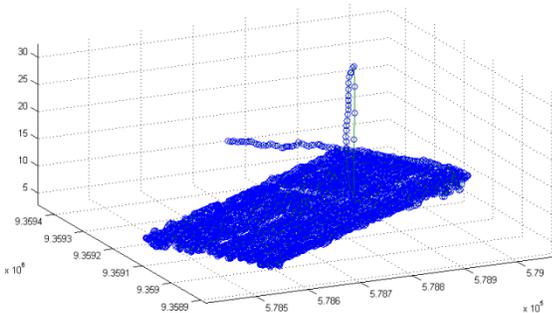


Fig. 2. Three-dimensionnal visualization of the navigation data. LAPMv2 provides options for visualizing, filtering and smoothing the navigation data.

(USBL) or long baseline (LBL), which generate geographic coordinates. This makes the preparation of the input files more straightforward. For calculations that require the use of projected units, LAPMv2 internally converts the geographic coordinates into either Universal Transverse Mercator (UTM) or Universal Polar Stereographic (UPS). The most appropriate of the two projections, as well as the relevant zone are automatically determined from the geographic coordinates. The UTM system is used for latitudes between 80°S and 84°N. Beyond these latitudes, the UPS system is automatically applied, thus ensuring coverage up to the poles. However, in case the mosaic area overlaps on several UTM or UPS zones, the zone of the most southwesterly image of the survey is used for the internal calculations throughout the mosaic.

#### B. Outputs of LAPMv2

All mosaic files produced by LAPMv2 are geo-referenced (Geotiff format) and directly importable into geographic information systems (GIS) for further analyses. As for the input data, the geo-referencing data of the output files use the WGS84 Geographic Coordinate System. Photomosaics can be produced in two different ways: *mosaic image* and *mosaic dataset*.

- The *mosaic image* option saves the entire mosaic as one single image file. In case of memory constraints, large mosaics can optionally be tiled and saved in several files of smaller size. The advantage of this option is that mosaic tiles can additionally be opened with any image processing program for rapid viewing of the mosaic. However, the construction of mosaic tiles is at higher computing costs.
- The *mosaic dataset* option is considerably faster in that a mosaic file *per se* is not generated. Instead, individual input images are given geographic attributes and saved in the Geotiff format. Individual images can then be imported into GIS in order to display the mosaic.

LAPMv2 also provides an extension toolbar for ArcGIS 10.2. This toolbar facilitates and batches the import of large datasets (hundreds to thousands of images) into ArcGIS. In particular, it ensures that all images of a *mosaic dataset* are correctly loaded into an ArcGIS file geodatabase and that the overviews are computed for several resolutions to enable faster display of the mosaic.

## IV. ALGORITHMS

#### A. Feature detection and matching

Image registration is primarily based on feature matching between any pairs of overlapping images. Although LAPMv2 offers the possibility to match features manually, most features are detected and matched automatically using the VLFeat implementation [13] of the scale-invariant feature transform (SIFT) algorithm [1], [2]. SIFT descriptors have proven most efficient for use with underwater imagery (low contrasts, few features, strong parallax distortions) and are currently

considered as the state-of-the-art method for such applications [14].

The computation of SIFT descriptors is, however, computationally expensive, and matching high-resolution images can be a slow process. This rapidly becomes an issue with datasets containing thousands of images. In order to speed-up the process, LAPMv2 gives the possibility to down-sample the image resolution during the feature detection and matching step. Tests showed a significant improvement in terms of computation time, with an unchanged cost in terms of registration accuracy (Fig. 3). In certain cases, down-sampling images prior to the feature detection leads to a larger *valid matches / total matches* ratio and, hence, to a better matching.

### B. Robust outlier rejection

Because the linear global registration is sensitive to false matches, it is extremely important to ensure that all outliers are removed beforehand. Such check can be performed by visualizing the matches between each pair of linked images, and erroneous matches (outliers) can be deleted manually

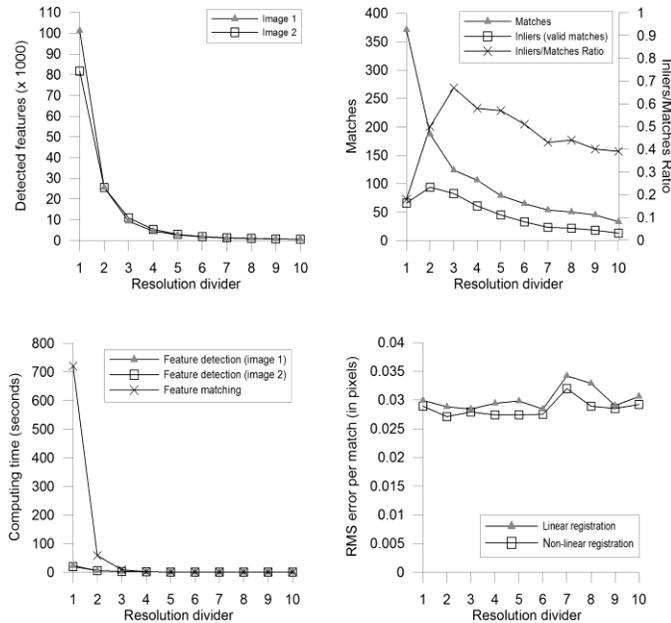


Fig. 3. Influence of the image resolution on the speed and performance of the feature detection and matching of the images shown in Fig. 4. *Top left:* the amount of detected features decreases sharply with the resolution; *Top right:* the amounts of detected matches (total and valid) decrease with the resolution; surprisingly, in this case the amount of valid matches (inliers) that are detected is higher at half-resolution than at full resolution; this is caused by the high proportion of false matches at full resolution, which affects the efficiency of the outlier rejection; however, the chosen images constitute a challenging dataset as they are affected by strong high-frequency grain, and such behavior is not observed with all datasets. *Bottom left:* dividing the image resolution by 2 led to a reduction of the matching time (including outlier rejection) by a factor of 12; reducing the resolution further did not significantly improve the matching time further. *Bottom right:* the image resolution does not appear to affect the average RMS error of the matches; note that the non-linear registration does not significantly reduce the RMS error further.

through the graphical interface. However, considering that large-area mosaicking commonly involves hundreds to thousands of images, searching for outliers in such large datasets manually will be extremely time consuming. Using an automatic outlier rejection, can spare the user such tedious work and time. Nevertheless, making the mosaicking process entirely autonomous without the need for any user supervision requires the outlier rejection algorithm to be extremely robust.

Random Sample Consensus (RANSAC) methods [15] alone are very efficient but they do not ensure removal of every erroneous match, in particular when the *inlier/total matches* ratio is small. LAPMv2 uses a somewhat more aggressive method in the sense that it removes too many rather than too few matches. In this implementation, the outlier rejection is

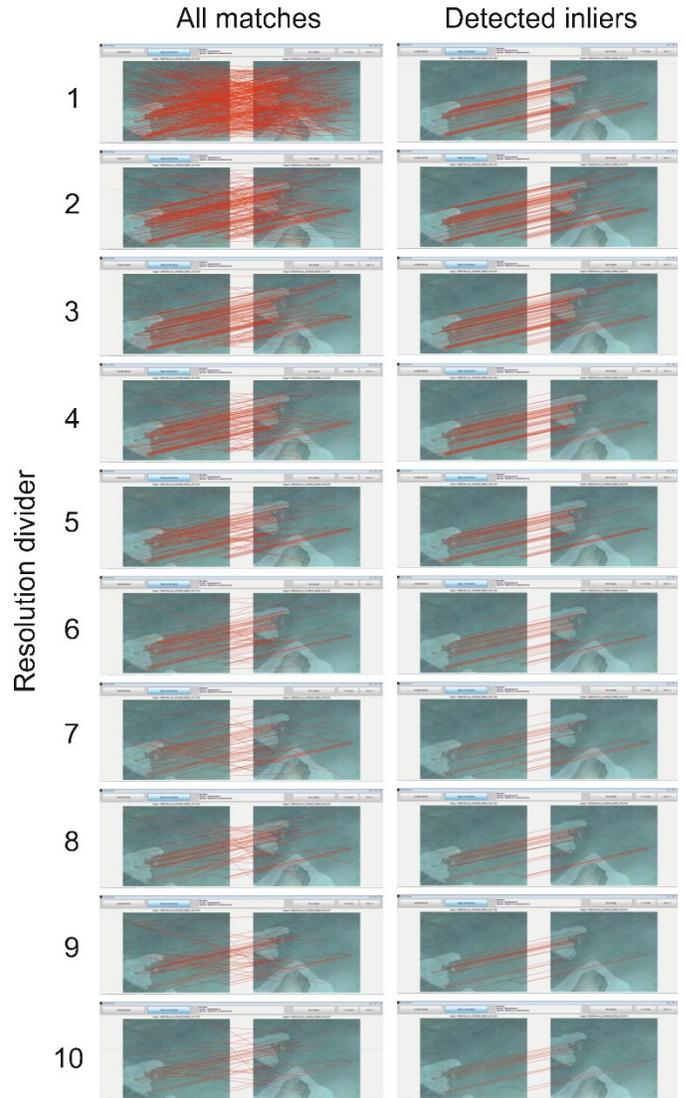


Fig. 4. Total and valid matches (inliers) detected by LAPMv2 at different image resolutions. A resolution divider of 1 means full resolution. Note that the high amount of erroneous matches at full resolution affected the performance of the outlier rejection step; this behavior was caused by a high-frequency grain noise in the full resolution images (4416x3312 pixels).

done in three steps:

- a RANSAC with homography estimation,
- a filtering by correlating the distances between the matched features of each images; in this step, the less correlated matches are removed iteratively until the correlation of the distances between matched features of both images satisfies a user-defined threshold,
- a filtering based on deviation estimation (see below); during this step, the most *deviating* matches are iteratively filtered out until the set threshold for deviation error becomes satisfied.

The deviation error of a match  $i$  in relation to a match  $j$  is calculated as follows:

$$D_{ij} = \left| \frac{A_{ij}^2}{\max(A|B)^2} - \frac{B_{ij}^2}{\max(A|B)^2} \times \left( \frac{|\widetilde{A}|}{|\widetilde{B}|} \right)^2 \right| \quad (1)$$

where  $A$  is the  $n$ -by- $n$  array of distances between all  $n$  matched features of image 1,  $B$  is the  $n$ -by- $n$  array of distances between all  $n$  matched features of image 2, and  $|\widetilde{A}|$  represents the median absolute values of array  $A$ . The ratio of the medians is used to scale the values of  $B$ .

The deviation array  $D$  is in turn used to score all matches based on the user-defined deviation threshold  $Dt$ , as in (2). The *most deviating* match is the one with the highest score  $S_i$ .

$$S_i = \sum_j [D_{ij} > Dt] \quad (2)$$

where the square brackets represent the conditional Iverson brackets. During the testing of the outlier rejection, good results were achieved with  $Dt$  set to  $2 \times 10^{-2}$ , and it has been kept as default value. However, the value of  $Dt$  can be modified by the user.

### C. Global registration

In mosaicking terminology, the *global registration* (or *bundle adjustment*) refers to the simultaneous estimation of the transformation parameters that map all images onto the mosaic plane with the lowest global error. This is done by minimizing a cost function, i.e. a mathematical formulation of the global registration error.

LAPMv2 can use both a linear and a nonlinear method to solve the minimization problem and compute the global registration. In both cases, the cost function used for the global registration is a modified version of the least-square problem presented in [3]. The advantage of this formulation is that it can be solved for affine transforms (6 degrees of freedom) by linear least squares and, thus, quickly lead to the global minimum solution. The resulting affine global solution can then be used as starting state for a non-linear minimization in order to further refine the global registration. The non-linear registration method used in LAPMv2 solves for planar-projective transforms (8 degrees of freedom) using the Levenberg-Marquardt algorithm [3], [16]–[19].

The cost term associated to the registration of matched images is defined, as in [3], by the sum of distances between corresponding features on the 2D frame of the mosaic. Using a similar notation format for describing transformation matrices as in [9], this cost term can be written as in (3).

$$C_{match} = \sum_{k,l} \sum_i \left| {}^m H \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} - {}^l H \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \right| \quad (3)$$

where  $k, l$  refers to a pair of matched images,  ${}^m H$  is the planar transformation from the image frame  $k$  to the 2D mosaic frame  $m$ , and  ${}^k(u_i, v_i, 1)^T$  and  ${}^l(u_i, v_i, 1)^T$  are the respective coordinates of the  $i$ -th match expressed in the 2D frames of images  $k$  and  $l$ .

The cost function used in LAPMv2 differs from that in [3], in that it permits the use of additional positioning constraints, referred to as *fiducial points* and *fiducial pairs*. The term *fiducial points* was introduced by [9] and [10], and refers to points on the seafloor with known absolute coordinates, such as landmarks or markers. Fiducial points can be used in the cost function as additional constraints to ensure that the given points will be registered correctly in the geo-referenced mosaic.

The cost associated to fiducial points is expressed, as in

$$C_{points} = \sum_{k=1}^N \sum_p \left| \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} - {}^m H \cdot \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \right| \quad (4)$$

where  $N$  is the total amount of images,  $p$  is a fiducial point related to image  $k$ ,  ${}^m(x_p, y_p, 1)^T$  are the known coordinates of the fiducial point  $p$  in the 2D mosaic plane  $m$ , and  ${}^k(u_p, v_p, 1)^T$  is the position of  $p$  in the image frame  $k$ .

Extending the terminology further, LAPMv2 introduces the concept of *fiducial pairs*. A *fiducial pair* is a pair of non-overlapping images whose relative position with each other is known. If fiducial pairs are provided, the relative position of the centers of the two images will be added as constraints to the cost function. The concept of fiducial pairs was initially added to prevent unbound growth of registration errors in mosaic loose ends during the linear registration. However, it turned out to be an efficient way of *fixing* the position of an image (for which the absolute coordinates are unknown) in relation to other images of the mosaic, in order to avoid aberrant solutions of the linear registration in poorly constrained areas of a mosaic. For instance, this capability is very useful to prevent adjacent but non-overlapping lines of mosaics from crossing each other due to registration errors adding up.

The cost associated to fiducial pairs is expressed, as in

$$C_{pairs} = \sum_{k1, k2} \left| \left( \begin{pmatrix} u_{c1} \\ v_{c1} \\ 1 \end{pmatrix} - {}^{k2} H \cdot \begin{pmatrix} u_{c2} \\ v_{c2} \\ 1 \end{pmatrix} \right) - \left( \begin{pmatrix} x_{c1} \\ y_{c1} \\ 1 \end{pmatrix} - \begin{pmatrix} x_{c2} \\ y_{c2} \\ 1 \end{pmatrix} \right) \right| \quad (5)$$

where  $k1, k2$  defines a fiducial pair, and  ${}^{k1}(u_{c1}, v_{c1}, 1)^T$  is the center of image  $k1$  expressed in the 2D frame of image  $k1$ .

Finally, in contrast with [3], the LAPMv2 implementation assumes that overlaps does not necessarily exist, i.e. that matching may not have succeeded, between all temporally consecutive images of the dataset. Indeed, in the absence of successful matching (either automatic or manual) between two consecutive images, the relative position of both images is estimated from the navigation data. In the cost functions, the navigation data is implemented in the form of *dummy matches*.

*Dummy matches* between images  $k$  and  $l$  are created by expressing the coordinates of the corner points of image  $k$  with respect to the 2D frame of image  $l$ , and vice versa. This is done by estimating the camera pose for each image from the navigation data (assuming a projective pinhole camera) and computing the corresponding planar transformations  $P$  from the image frames to the 2D mosaic frame [9].

The coordinates of the projection of a point  $i$  of image  $k$  onto the 2D mosaic plane  $m$  can be expressed, as in

$${}^m \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = {}^m P \cdot {}^k \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \quad (6)$$

where  ${}^m P$  is the planar transformation from the image frame  $k$  to the 2D mosaic frame  $m$  estimated from the navigation data,  ${}^k(u_i, v_i, 1)^T$  are the coordinates of the  $i$ -th feature expressed in the 2D frame of image  $k$ , and  ${}^m(x_i, y_i, 1)^T$  are the coordinates corresponding to the projection of the  $i$ -th feature onto the 2D mosaic frame  $m$ .

In a similar way, the point  ${}^m(x_i, y_i, 1)^T$  can then be expressed with respect to the 2D frame of image  $l$ , as in (7).

$${}^l \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = {}^m P^{-1} \cdot {}^m \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad (7)$$

Finally, expressions (6) and (7) can be combined to express the position of the  $i$ -th feature of image  $k$  with respect to the 2D frame of image  $l$ .

$${}^l \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = {}^m P^{-1} \cdot {}^m P \cdot {}^k \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \quad (8)$$

The cost of the dummy matches is calculated in a similar way as for the regular matches, as in

$$C_{nav} = \sum_i \left| {}^m P \cdot {}^k \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} - {}^m P \cdot {}^l \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \right| + \sum_j \left| {}^m P \cdot {}^k \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} - {}^m P \cdot {}^l \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \right| \quad (9)$$

where  $i$  and  $j$  features are corners points of images  $k$  and  $l$  respectively.

The global cost function has the form

$$C = C_{match} + \omega_{nav} \cdot C_{nav} + \omega_d \cdot C_d + \omega_p \cdot (C_{points} + C_{pairs} + C_0) \quad (10)$$

where  $C_d$  and  $C_0$  are the respective constraints for image distortion and overall translation of the mosaic [3], and  $\omega_{nav}$ ,  $\omega_d$  and  $\omega_p$  are weighting coefficients for the cost terms. The  $C_0$  cost term is only used if fiducial points are not provided.

The global registration is done by minimizing expression (10) over the parameters of the transforms of all  $N$  images, i.e. over the parameters of  ${}^m H \forall k \in [1, N]$ . The minimization problem can be solved either linearly (affine transforms) or non-linearly (planar projective transforms).

#### D. Mosaic rendering

LAPMv2 allows the user to choose between several rendering options (Fig. 5): no blending (Voronoi tessellation), maximum intensities, weighted mean and multi-frequency-bands blending.

- *No blending*: this rendering option consists of creating a Voronoi tessellation of the mosaic, i.e. a mosaic in which for each pixel is only shown the pixel intensity of the image whose center is the closest to that pixel [20].
- *Maximum intensities*: with this option, the mosaic is rendered according to the highest intensity of all overlapping pixels, i.e. the brightest pixels are selected. This option is a quick way of reducing vignetting effects in the mosaic. However, small registration errors may cause duplicated features or fuzziness in overlap areas.
- *Weighted mean*: this option consists of averaging pixels intensities of overlapping pixels using weighting factors that are inversely proportional to the distances to the center of the respective images. In other words, the closer a pixel is to the center of its original image, the more weight its intensity will have in the calculation of the average. This method is efficient at fading most seams but the mosaic may lose sharpness.
- *Multi-frequency-bands blending*: also known as multi-resolution blending [3], [6], [21], this method consists of decomposing the images into a set of frequency bands and to blend each band over a different spatial range. The high frequencies are blended over a small range, while the low frequencies are blended over a large range [21]–[23]. The multi-frequency blending implementation used by LAPMv2 is based on a Voronoi tessellation of the mosaic. This method is computationally expensive and much slower than the other rendering options, but it returns better results with less visible seams.

## V. RESULTS AND DISCUSSION

All functionalities have been successfully tested on eight datasets of different sizes (from 16 to 34,118 images), which resulted from cruise MSM15/2 in 2010 (RV Maria S. Merian) with the MARUM ROV Quest 4000 (Fig. 5), cruise LOOME in 2010 (RV Maria S. Merian) with the WHOI AUV Sentry (Fig. 6-7), and cruise WACS in 2011 (RV Pourquoi Pas?) with the IFREMER ROV Victor 6000 [24]–[26].

As shown in Fig. 3 and Fig. 4, down-sampling the input images during the automatic detection and matching phase results in a significant gain of time with no accuracy loss in the final registration. This is especially useful for the sidelap search in very large image datasets. For instance, searching for sidelap in a dataset of 34,118 images with search radii of 5 m and 10 m resulted in checking respectively 194,134 and 797,686 pairs of non-consecutive images for possible overlap. Although the precise duration of this mosaicking project has not been measured, if we consider that the automatic matching requires about 15 seconds per pair of images, a full search for sidelap would then have required approximately 1 month (5 m radius) and 4.5 months (10 m radius) respectively. Considering a similar time reduction factor as in the example shown in Fig. 3, the time required for a full sidelap search would drop to less than 3 and 12 days respectively. Although the gain of time is significant, the timescales involved remain noticeable.

In that regard, the advantages of running the whole process in an unsupervised and autonomous fashion are obvious. Although robust, the false-match rejection of the former LAPM Tool [11] was not fully foolproof, and matched pairs had to be visually checked by the user in order to ensure that all erroneous matches were removed prior to running the global registration step. It is evident that such visual inspection effort is, especially for datasets as large as the above-mentioned one, not only extremely time-consuming, but also tedious and ungratifying work.

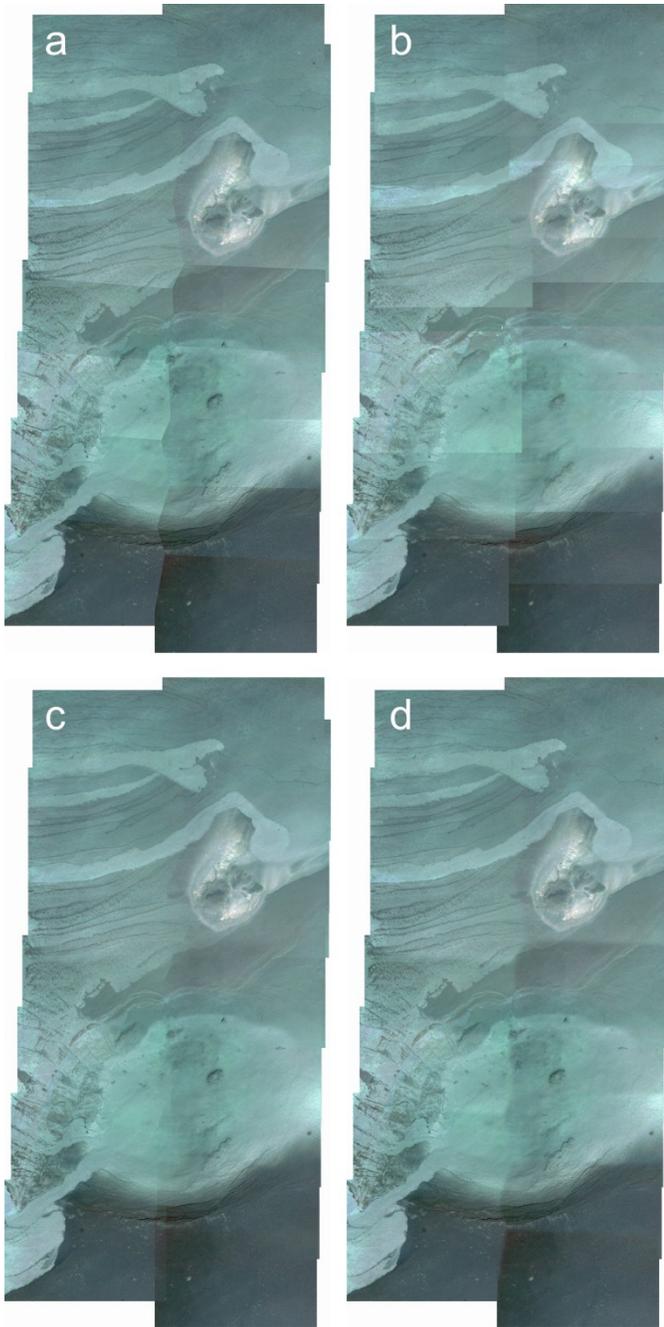


Fig. 5. Rendering options of LAPMv2. (a) *No blending*: the seams between the images (i.e. limits of the Voronoi regions) are clearly visible. (b) *Maximum intensities*: the mosaic is generally brighter but seams are still visible; duplicated features or fuzziness occur. (c) *Weighted mean*: the seams are barely visible; however, the mosaic loses sharpness due to averaging of small registration errors. (d) *Multi-frequency-bands blending*: some seams are slightly more visible than with the weighted mean method, but the mosaic is sharper. The photomosaic is made out of 16 photographs taken with the ROV Quest 4000 during cruise MSM15/2 to the Black Sea. It covers an area of about 50 m<sup>2</sup> and shows mud flows and mud pools at the Helgoland mud volcano. The photographs were pre-processed in order to enhance the contrasts and colours, and to reduce the lighting heterogeneities.

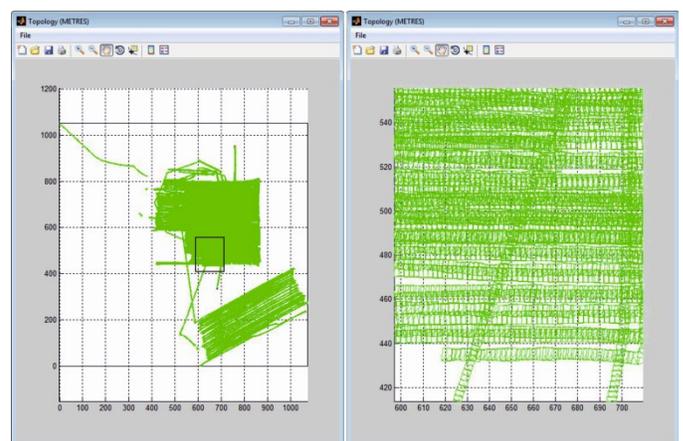


Fig. 6. Image alignment (topology) of the LOOME photomosaic of Håkon Mosby Mud Volcano (34,118 photos) after the linear global registration. *Left*: overview of the entire mosaic area. *Right*: close-up on the area indicated by the black rectangle, on which the individual image frames are discernible. The units are in meters. The display of the image frames is inspired from that produced by the tool presented in [10].

The outlier rejection method presented in this work was tested on several datasets containing up to 34,118 images. In the latter dataset, visual inspection of all consecutive and non-consecutive matched pairs (46,090 matched pairs) did not reveal the presence of a single erroneous match. This outcome confirmed results from prior testing on smaller datasets and, hence, greatly heightened the confidence in the false-match rejection algorithm.

Unsurprisingly, the amount of valid matches per pair is generally lower with LAPMv2 than with the LAPM Tool (66 against 78 for the example shown in Fig. 3). Yet, the difference is not only due to the presence of erroneous matches in the results of the outlier rejection of the former tool. Indeed, part of the difference seems to be caused by the fact that the outlier rejection algorithm of LAPMv2 is generally more aggressive than its former counterpart and, thus, it tends to filter out more matches than is strictly necessary. However, these results strongly suggest that LAPMv2 is suitable for fully unsupervised image matching and, hence, adequate for autonomous mosaicking. Such capability results in an important gain of time for the user and significantly speeds up the mosaicking process. Nevertheless, users also have the possibility to manually edit and create matches in order to link overlapping images in case the automatic matching fails. This will then result in a more accurate alignment of the images than would have been obtained had the navigation data been used instead (default behavior for aligning unmatched images).

Apart from the differences described in Section IV, the global registration of LAPMv2 is similar to that of previous works [3], [11], [22]. In areas with no to low relief, the produced photomosaics show that the overlap areas are accurately registered. This is indicated, for instance, by the continuity of linear features, which are well rendered with no major breaks, as in Fig. 5. Minor registration errors are still visible in large mosaics, which are mostly caused by the non-planarity of the scenes. However, where present, those errors are in the order of a few centimeters ( $< 10$  cm) only, which is below the precision required by most large scale spatial analyses (e.g. hundreds of square meters or more). For example, results and conclusions of mosaic-based analyses of megafaunal distribution will not be affected by such errors [26]–[28]. By contrast, in areas of strong relief variations, the mosaics may be impacted by larger registration errors or show ill-rendered structures. As mentioned above, performances of 2D imaging techniques in such cases are poor, and it is up to the users to knowledgeably analyze and interpret the mosaics.

Comparing the quality of the global registration between different mosaicking techniques in a quantitative way is difficult. One common criterion is to compute the RMS error of the distances between corresponding matches in the mosaic. Although this method may be suitable for assessing the registration quality of a pair of carefully-chosen images as in Fig. 3, the RMS error is not a measure of the global consistency or correctness of the mosaic. Indeed, all 2D mosaicking techniques are based on the same assumption that the scene is planar. In practice, this assumption is never fully satisfied and registration errors will occur. In such context, the RMS error of the global mosaic cannot reach a null value without introducing anomalous distortions in some images.

Therefore, imaging a 3D scene onto a 2D plane involves a tradeoff between level of distortion of the images and low RMS error of the corresponding matches in the mosaic.

In LAPMv2, this tradeoff is controlled by the user through the use of the weighting coefficient ( $\omega_d$ ) for the distortion term of the cost function. The correctness of the mosaic can then be qualitatively assessed by visual inspection or, for instance, by

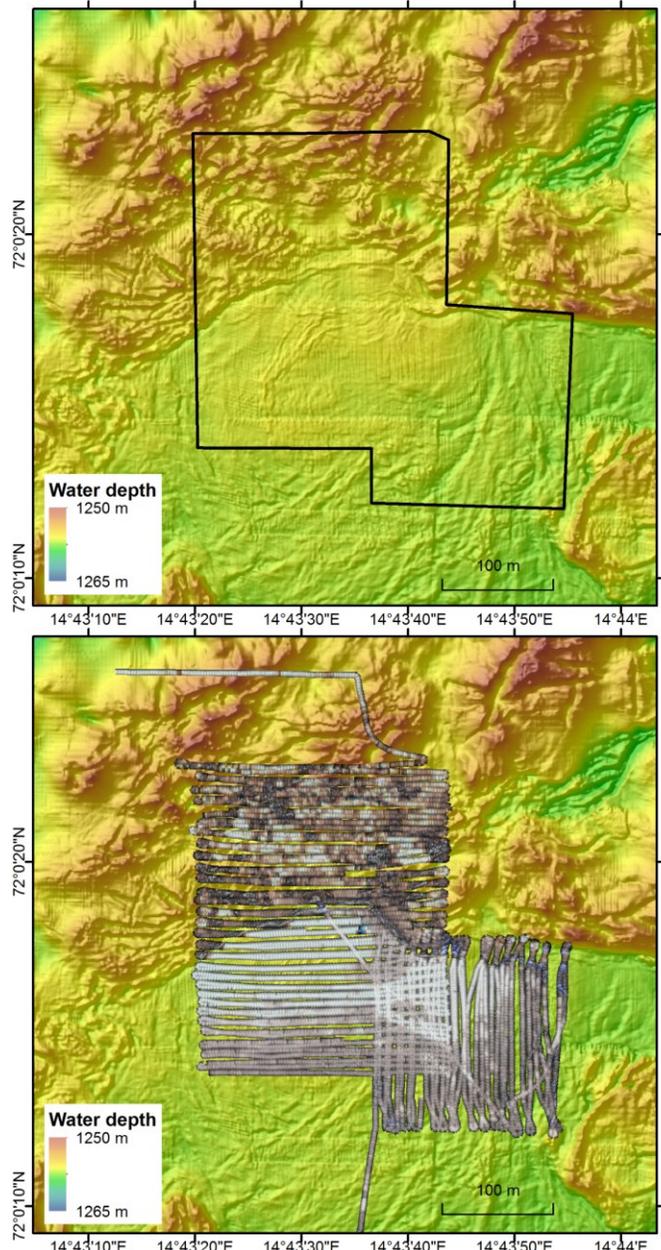


Fig. 7. Subset of the photomosaic of the Håkon Mosby Mud Volcano made of 8,083 images (out of the 34,118 images of the entire dataset). The area delineated by the black box covers approximately 86,700 m<sup>2</sup> (calculated using a Mollweide equal-area projection to eliminate distortions due to the high latitude). The main pattern visible on the photomosaic matches perfectly with the main structures of the underlying micro-bathymetry, which confirms that the photomosaic is accurately georeferenced.

comparing the mosaic with micro-bathymetry data of the same area (Fig. 7). Indeed, micro-bathymetry data can be used to identify zones with rugged topography, i.e. zones where the mosaic may be impacted by the largest errors. This way, in particular in flat areas or areas with low and gradual relief variations, 2D mosaicking techniques proved efficient at mapping large areas with an accuracy that permitted spatial analyses to be performed [10], [29], [26].

In comparison to its predecessor, LAPMv2 is not only considerably faster and autonomous, but it is also more robust, and has more functionalities as well as more flexibility in the workflow, the input and output formats, and the geo-referencing method [11]. For instance, the current implementation uses navigation data to assess the camera poses as well as the motion between unmatched images, which allows a better conditioning of the dummy matches. Therefore, this implementation is more consistent in the formulation of the cost function for the global registration, and leads to more accuracy in the final registration than in previous versions.

Furthermore, the *mosaic dataset* output function of LAPMv2 gives the possibility to geo-reference and save each individual image as a Geotiff file. It enables a rapid construction of the mosaic but requires loading every image as a separate raster dataset into GIS. Depending on the computer system or on the GIS that are used, this process may hang or fail. The LAPM Toolbar for ArcGIS prevents such problem and ensures that the import process completes properly. Nevertheless, display issues were observed when displaying the largest test dataset (34,118 images) under ArcGIS. This behavior was likely due to the enormous amount of images causing instabilities in the GIS program, and the fix to that problem was not straightforward. In future versions, a workaround may involve using the mosaic viewing program developed by the Computer Vision and Robotics Group at the University of Girona and presented in [10].

## VI. CONCLUSION AND FUTURE WORK

The LAPMv2 tool does not claim to compete with all existing photo-mosaicking techniques. Instead, it was designed to provide marine scientists with an intuitive and out-of-the-box tool for creating underwater photomosaics. To that end, it is freely available to end-user scientists. The current version of the program is LAPM v2.3.5. It is both robust and fully functional, but there is still clear room for improvements. Therefore, LAPMv2 will be subject to ongoing development, and future versions will include the following improvements:

- Computation of SIFTs over graphical processing unit (GPU) instead of central processing unit (CPU) for faster feature detection and matching.
- Compensation for radial distortion.
- Refined navigation filtering and smoothing based on and extended Kalman filter.
- C implementation of the multi-band blending for faster computation of the mosaic.

- Multi-band blending based on a different seaming method than the Voronoi tessellation.
- Compatibility of the LAPMv2 output files with the *MosaicViewer* viewing program of the Computer Vision and Robotics Group of the University of Girona (<http://eia.udg.es/~rafa/mosaicviewer.html>).

## ACKNOWLEDGMENT

The author would like to thank Antje Boetius (Alfred Wegener Institute), Dirk de Beer (Max Plank Institute) and Dana Yoerger (Woods Hole Oceanographic Institute) for providing the large image dataset of the Håkon Mosby Mud Volcano. The author would also like to acknowledge the chief scientists of cruises MSM15/2, MSM16/2 and WACS, the captains and crews of RV Maria S. Merian and RV Pourquoi Pas?, as well as the teams of AUV Sentry, of ROV Quest 4000, and of ROV Victor 6000 for their contribution in producing the image datasets presented in this work or used for the development and testing of LAPMv2.

## REFERENCES

- [1] D. G. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*, Corfu, Greece, 1999, vol. 2, pp. 1150–1157.
- [2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [3] O. Pizarro and H. Singh, "Toward large-area mosaicking for underwater scientific applications," *IEEE J. Ocean. Eng.*, vol. 28, no. 4, pp. 651–672, Oct. 2003.
- [4] N. Gracias and J. Santos-Victor, "Underwater video mosaics as visual navigation maps," *Comput. Vis. Image Underst.*, vol. 79, no. 1, pp. 66–91, Jul. 2000.
- [5] N. Gracias and J. Santos-Victor, "Underwater mosaicking and trajectory reconstruction using global alignment," in *Proceedings of the MTS/IEEE Oceans 2001 Conference*, Honolulu, Hawaii, USA, 2001, p. 7 pp.
- [6] R. Eustice, O. Pizarro, H. Singh, and J. Howland, "UWIT: Underwater Image Toolbox for optical image processing and mosaicking in MATLAB," in *Proceedings of the 2002 International Symposium on Underwater Technology, 2002.*, Tokyo, Japan, 2002, pp. 141–145.
- [7] A. G. Vincent, N. Pessel, M. Borgetto, J. Jouffroy, J. Opderbecke, and V. Rigaud, "Real-time geo-referenced video mosaicking with the MATISSE system," in *Proceedings of the MTS/IEEE Oceans 2003 Conference*, San Diego, USA, 2003, vol. 4, pp. 2319–2324.
- [8] A. G. Allais, M. Borgetto, J. Opderbecke, N. Pessel, and V. Rigaud, "Seabed video mosaicking with MATISSE: a technical overview and cruise results," in *Proceedings of The Fourteenth International Offshore and Polar Engineering Conference*, Toulon, France, 2004.
- [9] J. Ferrer, A. Elibol, O. Delaunoy, N. Gracias, and R. Garcia, "Large-area photo-mosaics using global alignment and navigation data," in *Proceedings of the MTS/IEEE Oceans 2007 Conference*, Vancouver, British Columbia, Canada, 2007, pp. 1–9.
- [10] J. Escartín, R. García, O. Delaunoy, J. Ferrer, N. Gracias, A. Elibol, X. Cufi, L. Neumann, D. J. Fornari, S. E. Humphris, and J. Renard, "Globally aligned photomosaic of the Lucky Strike hydrothermal vent field (Mid-Atlantic Ridge, 37°18.5'N): Release of georeferenced data, mosaic construction, and viewing software," *Geochem. Geophys. Geosystems*, vol. 9, no. 12, p. 17 pp., Dec. 2008.
- [11] Y. Marcon, H. Sahling, and G. Bohrmann, "LAPM: a tool for underwater large-area photo-mosaicking," *Geosci. Instrum. Methods Data Syst.*, vol. 2, no. 2, pp. 189–198, Jul. 2013.
- [12] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," *www.vlfeat.org*, 2008. [Online]. Available: <http://www.vlfeat.org/>. [Accessed: 27-Jun-2014].

- [13] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proceedings of the international conference on Multimedia*, Firenze, Italy, 2010, pp. 1469–1472.
- [14] C. J. Prabhaka and P. U. Praveen Kumar, "Feature Tracking of Objects in Underwater Video Sequences," *ACEEE Int. J. Inf. Technol.*, vol. 2, no. 1, pp. 42–46, Mar. 2012.
- [15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [16] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Q. J. Appl. Mathematics*, vol. II, no. 2, pp. 164–168, 1944.
- [17] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.
- [18] D. P. Bertsekas, *Nonlinear Programming: 2nd Edition*, 2nd edition (September 1, 1999). Athena Scientific, 1999.
- [19] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, G. A. Watson, Ed. Springer Berlin Heidelberg, 1978, pp. 105–116.
- [20] D. Lirman, N. Gracias, B. R. Gintert, A. C. . Gleason, R. P. Reid, S. Negahdaripour, and P. Kramer, "Development and application of a video-mosaic survey technology to document the status of coral reef communities," *Environ. Monit. Assess.*, vol. 125, no. 1–3, pp. 59–73, Aug. 2007.
- [21] 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.
- [22] S. Hsu, H. S. Sawhney, and R. Kumar, "Automated mosaics via topology inference," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 44–54, Mar. 2002.
- [23] M. Brown and D. G. Lowe, "Recognising panoramas," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings, 2003*, pp. 1218–1225 vol.2.
- [24] A. Boetius, D. de Beer, J. Felden, M. Viehweger, V. Asendorf, E. Weiz, R. Stiens, G. Eicken, G. Wetzel, F. Harmegnies, L. Dussud, B. Ferré, J. Van Campenhout, S. Albrecht, C. R. German, R. Camilli, A. Billings, J. C. Kinsey, A. Duester, D. R. Yoerger, W. Versteeg, J. Verduyck, and K. de Rycker, "Short Cruise Report - MERIAN MSM16/2 - 'LOOME'-Long Term Observatory of Mud Volcano Eruptions," Alfred Wegener Institut für Polar und Meeresforschung, Bremerhaven, Cruise report, 2010.
- [25] G. Bohrmann, V. Blinova, K. Dehning, D. Evtushenko, C. Friese, A. Hiruta, D. Hüttich, M. Ivanov, S. A. Klapp, J. H. Körber, G. Komakhidze, E. Kopsiske, K. Lange, H. A. Mai, T. Malakhova, Y. Marcon, G. Meinecke, T. Pape, V. Ratmeyer, R. Rehage, J. Renken, C. Reuter, M. Reuter, M. Römer, H. Sahling, E. Sakvarelidze, P. Wintersteller, and M. Zarrouk, "Report and preliminary results of RV MARIA S. MERIAN Cruise MSM 15/2, Istanbul (Turkey) - Piraeus (Greece), 10 May - 2 June 2010. Origin and structure of methane, gas hydrates and fluid flows in the Black Sea," Geowissenschaften, Universität Bremen, Bremen, Cruise report 278, 2011.
- [26] Y. Marcon, H. Ondréas, H. Sahling, G. Bohrmann, and K. Olu, "Fluid flow regimes and growth of a giant pockmark," *Geology*, vol. 42, no. 1, pp. 63–66, Jan. 2014.
- [27] K. Olu-Le Roy, J. C. Caprais, A. Fifis, M. C. Fabri, J. Galéron, H. Budzinsky, K. Le Ménach, A. Khripounoff, H. Ondréas, and M. Sibuet, "Cold-seep assemblages on a giant pockmark off West Africa: spatial patterns and environmental control," *Mar. Ecol.*, vol. 28, no. 1, pp. 115–130, 2007.
- [28] Y. Marcon, H. Sahling, A. G. Allais, G. Bohrmann, and K. Olu, "Distribution and temporal variation of mega-fauna at the Regab pockmark (Northern Congo Fan), based on a comparison of videomosaics and geographic information systems analyses," *Mar. Ecol.*, vol. 35, no. 1, pp. 77–95, 2014.
- [29] T. Barreyre, J. Escartin, R. Garcia, M. Cannat, E. Mittelstaedt, and R. Prados, "Structure, temporal evolution, and heat flux estimates from the Lucky Strike deep-sea hydrothermal field derived from seafloor image mosaics," *Geochem. Geophys. Geosystems*, vol. 13, no. 4, p. 29 pp., Apr. 2012.