# Large-Area Photo-Mosaics Using Global Alignment and Navigation Data

J. Ferrer, A. Elibol, O. Delaunoy, N. Gracias and R. Garcia
Computer Vision and Robotics Group
University of Girona, 17071, Spain
{jferrerp, aelibol, delaunoy, ngracias, rafa}@eia.udg.es

***Abstract*—Seafloor imagery is a rich source of data for the study of biological and geological processes. Among several applications, still images of the ocean floor can be used to build image composites referred to as photo-mosaics. Photo-mosaics provide a wide-area visual representation of the benthos, and enable applications as diverse as geological surveys, mapping and detection of temporal changes in the morphology of biodiversity. We present an approach for creating globally aligned photo-mosaics using 3D position estimates provided by navigation sensors available in deep water surveys. Without image registration, such navigation data does not provide enough accuracy to produce useful composite images. Results from a challenging data set of the Lucky Strike vent field at the Mid Atlantic Ridge are reported.**

## I. Introduction

Image based mapping techniques provide unique opportunities to study benthic structures at locations not easily accessible to scientific divers. Such techniques, complemented with navigational data from positioning sensors have the potential to allow the accurate and repetitive mapping of study areas. Furthermore, a platform equipped with these sensing capabilities requires minimal end-user intervention during mission execution, thus benefiting a potentially large group of marine scientists. It is well known that the characteristics of the underwater environment offer several challenges, mainly due to the significant attenuation and scattering of visible light [1]. Moreover, light attenuation does not allow images to be taken from a large distance [2]–[4]. Therefore, in order to gain global perspective of the surveyed area, mosaicing techniques are needed to compose a large number of images into a single one. On the other hand, while seafloor imagery is commonly available, they are underexploited as seafloor mosaics are not routinely created.

Mosaicing techniques have been used to map areas of few hundreds of square meters while having enough ground resolution to allow the identification of colonies of a few centimeters size [5]. Over areas of low topographic relief, such techniques allow creating mosaics from image information alone [6]. However, when attempting to cover areas of more than 500 square meters or with high relief, the use of additional positioning information becomes crucial. Such information can be provided by acoustic positioning systems, inclinometers and rate gyros. Latest technological developments have significantly reduced the price and size of these sensors, thus allowing for their use even in small frame platforms [6], [7].

Recent impressive progress has been achieved in the field of simultaneous localization and mapping (SLAM) for underwater platforms equipped with cameras [8], [9] and sonars [10], [11]. SLAM approaches are well suited to navigation applications namely in the real-time positioning and control of vehicles. This contrasts with off-line batch approaches where the data is processed *a posteriori*. By avoiding real-time constraints, large scale methods and datasets can be optimized, with significantly higher accuracy in the final results. Our approach fits in this category, relating to a number of previous works.

One of the first mosaicing systems was proposed by Stanford/MBARI [12]. They developed a completely autonomous column-based mosaicking system by using a constrained four-parameter similarity motion model. Recently, this work was extended by Richmond and Rock who used a Doppler Velocity Log (DVL) to complement camera measurements [13]. Pizarro et al. [14] proposed a mosaicing system that exploited navigation and attitude information for bundle adjustment. Another mosaicing approach was reported by Madjidi and Negahdaripour [15], who solved the global alignment problem for a submersible equipped with stereo cameras. They proposed the use of mixed adjustment model to recursively determine the pose of the vehicle. Rzhanov et al. described in [16] a methodology that exploited navigation data to build geo-referenced photo-mosaics of the mid-ocean ridges at the East Pacific Rise. Some other approaches explicitly rely on absolute positioning sensors, Vincent et al. presented in [17] a software that integrates video mosaics in a geo-referenced environment. Their system takes profit of both acoustic USBL positioning and a set of dead-reckoning sensors.

### A. Objective and Motivation

This paper presents a technique to create photo-mosaics using the navigation data and the image set coming from typical seafloor surveys. The method is motivated by the need to create mosaics from large-area photo surveys when only sparse position and orientation information are available. This paper represents an effort in creating a set of mosaicing tools that will be made freely available for the marine science community. A central goal is to enable scientists to easily create large underwater mosaics, without requiring extended knowledge in image processing and optimization.

The structure of the paper is as follows: first, section II

0-933957-35-1 

describes how an estimate of the initial trajectory is generated from the navigation data. Then, this estimated trajectory is used to predict overlaps between consecutive images, and loops in the trajectory, as introduced in section III. Next, a bundle adjustment approach to optimally combine image registration information and navigation data is developed in section IV. Finally, section V shows the obtained results.

### B. Notation

In this paper we use a notation coming from robotics nomenclature [18] extended to take into account 3D and 2D elements:

- *3D/2D elements*: Names that refer to a 3D element (e.g. frames and vectors names) are written in capital letters. Non-capital letters are used to name 2D elements or scalars.
- *Frames*: Frame names are expressed within braces $\{\}$.
- *Vectors*: Leading superscript stands for the frame in which a vector is expressed. There is no specific convention for trailing subscript, but it usually identifies a vector component (e.g. $x$, $y$ and $z$) or an index in a list. For instance, given ${}^{M}T = {}^{M}(t_x, t_y, t_z)$ that is a translation $T$ in the frame $\{M\}$, the $x$ component is referred as ${}^{M}t_x$ to keep track from which frame and 3D vector it comes from and it is written using non capital $t$ because it is an scalar. The square of a vector is defined as the square of its components, e.g. ${}^{M}T^2 = {}^{M}(t_x^2, t_y^2, t_z^2)$. The operator $\langle\rangle$ stands for the dot product between two vectors, i.e. $\langle u, v\rangle = u_x \cdot v_x + u_y \cdot v_y$. The norm of a vector is written as $\|v\| = \sqrt{\langle v, v\rangle}$.
- *Transformation Matrices*: Leading subscript stands for the initial vector space and leading superscript for the final vector space. For instance, ${}^{W}_{M}R$ is a $3 \times 3$ rotation matrix than transforms coordinates from the $\{M\}$ frame to $\{W\}$ frame. Trailing superscript is used to denote the inverse or the transpose ($R^{-1}$, $R^{\top}$) and the trailing subscript is normally used as index in lists.
- *Homogeneous coordinates*: Names of vectors in 2D or 3D that are expressed in homogeneous coordinates are written with tilde. Then, we can write $T = (t_x, t_y, t_z)$, $\tilde{T} = (t_x, t_y, t_z, 1)$ or $t = (t_x, t_y)$, $\tilde{t} = (t_x, t_y, 1)$. Prior to any addition operation between a homogeneous vector and a non-homogeneous one, the scale must be removed. Therefore, we write $T = A + \tilde{B}$ meaning $T = (a_x + \frac{b_x}{b_s}, a_y + \frac{b_y}{b_s}, a_z + \frac{b_z}{b_s})$ if $A = (a_x, a_y, a_z)$ and $\tilde{B} = (b_x, b_y, b_z, b_s)$.

## II. Initial Estimation from Navigation Data

The camera is modeled as the classical projective pinhole camera [19, pp. 155-157] whose projection equation is

$$\tilde{x} = K \cdot R \cdot [I \mid -C] \tag{1}$$

where, $\tilde{x}$ is the up-to-scale imaged 2D point, $K$ is the intrinsic parameter matrix (2), $R$ is the rotation between the camera and the world frame, $I$ is a $3 \times 3$ identity matrix and $C$ is the camera translation with respect to the world coordinate frame.

$$K = \begin{pmatrix} \alpha_x & 0 & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Names in (1) are changed to agree with the notation presented in section I-B, resulting in (3):

$${}^{i}\tilde{x} = K \cdot {}^{C}_{W}R \cdot \left[I \mid -{}^{W}T\right] \cdot {}^{W}X \tag{3}$$

where ${}^{C}_{W}R$ is the $3 \times 3$ rotation matrix that transforms vector coordinates defined with respect to the world coordinate frame $\{W\}$ to the camera frame $\{C\}$, ${}^{W}T$ is the translation of the camera center with respect to the frame $\{W\}$ and ${}^{i}\tilde{x}$ is the 2D point in homogeneous coordinates that corresponds to the projection onto the image plane of the point ${}^{W}X$, defined with respect to the world frame.

The navigation data is assumed to contain the position and the heading of the vehicle with respect to the coordinate frame $\{W\}$ placed at the origin of the corresponding UTM zone. Since the surveyed area can be far from the origin of the UTM zone, large numbers for coordinates will lead to poor numerical conditioning in further processing. At the same time, we need a 2D mosaic frame $\{m\}$ to represent the resulting photo-mosaic image. For these reasons, a 3D mosaic frame $\{M\}$ is set up in a way that the 2D mosaic frame is the plane $z = 0$ with all the initial vehicle poses within the $x$-$y$ positive quadrant. The camera has its own coordinate frame $\{C_j\}$, related to the vehicle frame through a rigid motion transformation. This configuration is depicted in Fig. 1.

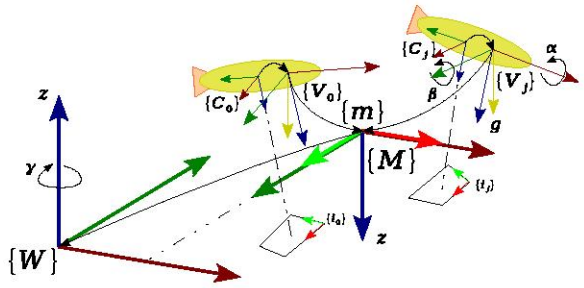



Fig. 1. Vehicle poses $\{V_j\}$ are defined with respect to a 3D mosaic frame $\{M\}$. There is a fixed rigid motion between the vehicle frame and the camera frame $\{C_j\}$ as well as between the world frame $\{W\}$ and the 3D mosaic frame $\{M\}$. Images $\{i_j\}$ taken at each camera pose are projections of the image plane onto the $z = 0$ plane in the 3D mosaic frame $\{M\}$ that corresponds to the 2D mosaic frame $\{m\}$. The angles $\alpha$, $\beta$ and $\gamma$ show the axis of rotation for roll, pitch and heading respectively. Yellow axis in the vehicle frames $\{V_0\}$ and $\{V_j\}$ represent the gravity vector that we define as antiparallel to the $z$ axis of the world frame $\{W\}$.

Let us define the surveyed area (plane $x$-$y$ in the frame world frame $\{W\}$) so that it is orthogonal to the gravity vector (yellow vector $g$ in Fig. 1). Therefore, gravity vector and $z$ axis of the world frame $\{W\}$ are antiparallel.

The rotation ${}^{W}_{V}R$ for each vehicle pose can be calculated using the navigation angular readings as it is shown in (4).

$$ {}^{W}_{V}R = R_x(\pi) \cdot R_z(\gamma - \pi/2) \cdot R_y(\beta) \cdot R_x(\alpha) \qquad (4) $$

where, $R_x$, $R_y$ and $R_z$ denote the rotation operators in the different axes [18, pp. 34]. The first rotation in $x$ axis aligns the $z$ axis of the world frame $\{W\}$ with the gravity vector. Then the heading rotation $\gamma$ around the $z$ axis is carried out. The $-\pi/2$ constant compensates for the discrepancy between heading that has the 0 on the North ($y$ axis in $\{W\}$) with respect to the 0 angle in the $\{W\}$ coordinate system ($x$ axis). Finally, pitch and roll rotations are accounted for as shown in Fig. 2. The vehicle's translation ${}^{W}T_v$ is defined directly from the position provided by the navigation data.

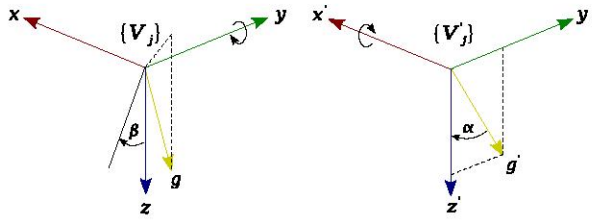



Fig. 2. The two pictures represent a vehicle frame $\{V_j\}$ that is deviated from the gravity vector $g$. A rotation in $y$ axis (left) projects the gravity vector $g$ to the plane $y$-$z$ resulting in $g'$ (right). The pitch angle ($\beta$) that can be seen as the angle between $z$ axis and the projection of $g$ onto the plane $x$-$z$. After the first rotation, we obtain a new frame $\{V'_j\}$ in which the roll ($\alpha$) is applied around $x$ axis to align the vector $g'$ with the $z$ axis.

So far, we have vehicle poses $({}^{W}_{V}R, {}^{W}T_v)_j$ defined in world frame. The aim now is to have camera poses with respect to the 3D mosaic frame. This can be achieved by carring out rigid motion operations that transform ${}^{W}_{V}R$ and ${}^{W}T_v$. Applying the motions ${}^{W}_{M}R$ and ${}^{C}_{V}R$ to ${}^{W}_{V}R$ we can obtain the rotation ${}^{C}_{M}R$, i.e. ${}^{C}_{M}R = ({}^{W}_{M}R^{-1} \cdot {}^{W}_{V}R \cdot {}^{C}_{V}R^{-1})^{-1}$. In the same way, the position of the vehicle ${}^{W}T_v$ with respect to the world frame can be transformed to the position of the camera ${}^{M}T = (t_x, t_y, t_z)^\top$ with respect to the mosaic frame. Then, the vehicle frames $\{V_j\}$ and the world frame $\{W\}$ are no longer used.

Expression (3) can be expanded to:

$$ {}^{i}\begin{pmatrix} \rho u \\ \rho v \\ \rho \end{pmatrix} = K \cdot {}^{C}_{M}R \cdot \begin{pmatrix} 1 & 0 & 0 & -{}^{M}t_x \\ 0 & 1 & 0 & -{}^{M}t_y \\ 0 & 0 & 1 & -{}^{M}t_z \end{pmatrix} \cdot {}^{M}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \qquad (5) $$

For each camera pose $j$ we have a rotation matrix ${}^{C}_{M}R_j$ and a translation vector ${}^{M}T_j = (t_x, t_y, t_z)^\top$.
From (5) we can define an *image*-to-*ground plane* mappings [20] by projecting to the mosaic plane $z = 0$ obtaining (6):

$$ {}^{i}\begin{pmatrix} \rho u \\ \rho v \\ \rho \end{pmatrix} = K \cdot {}^{C}_{M}R \cdot \begin{pmatrix} 1 & 0 & -{}^{M}t_x \\ 0 & 1 & -{}^{M}t_y \\ 0 & 0 & -{}^{M}t_z \end{pmatrix} \cdot {}^{M}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (6) $$

Finally, (7) represents the planar motion between each frame $i$ and the mosaic frame $\{m\}$.

$$ {}^{i}_{m}H = K \cdot {}^{C}_{M}R \cdot \begin{pmatrix} 1 & 0 & -{}^{M}t_x \\ 0 & 1 & -{}^{M}t_y \\ 0 & 0 & -{}^{M}t_z \end{pmatrix} \qquad (7) $$

Using the planar motion described by (7), a 2D point given with respect to mosaic frame $\{m\}$ (i.e. a 3D point defined with respect to the frame $\{M\}$ projected onto the plane ${}^{M}z = 0$) can be transformed to the image frame $i$. In the same way, ${}^{i}_{m}H^{-1} = {}^{m}_{i}H$ can be used to transform a point expressed with respect to the image frame $i$ to the 2D mosaic frame. Furthermore, the relative planar motion between overlapping frames $i$, $j$ can be computed by composing the absolute ones ${}^{i}_{j}H = {}^{i}_{m}H \cdot {}^{j}_{m}H^{-1} = {}^{i}_{m}H \cdot {}^{m}_{j}H$.

To summarize, as output of this section we have:

1) For each image frame $i$ we have computed its 3D camera pose, defined with respect to the mosaic frame $\{M\}$ using sensor data. These poses will be used as initial guess for bundle adjustment in section IV.
2) The planar transformation from the image frame to the mosaic frame ${}^{m}_{i}H$ can be used to build a (non-optimized) photo-mosaic from navigation data. Moreover, this planar transformation will be used in the following section to estimate the overlapping image pairs, prior to image matching.

## III. Overlap Detection and Image Matching

We define two different sets of candidate overlapping pairs of frames in the trajectory:

- *Sequential:* We assume that each pair of consecutive entries in the navigation data are potential candidates to be a sequential pair of overlapping frames (although they do not overlap in some cases).
- *Non-Sequential:* As the estimated trajectory is noisy and may have a drift, a method to include all the possible crossings has been devised. The center of each image is computed in the 2D mosaic frame according to planar transformations ${}^{m}_{i}H$ obtained in previous section. The sequence of images is explored to produce candidate pairs between the current image and all those that are closer than a certain threshold and that have a lower index [1].

The candidate pairs are matched using a robust matching procedure. This matching is performed in three steps. As a first step, SURF [21] features are extracted from both images. The features are matched using RANSAC [22] under a planar projective motion model (homography with 8 degrees of freedom) [23], [24]. As a second step, a number of verifications are performed to ensure that this homography corresponds to a valid camera motion:

[1]Images are identified by its position in the trajectory (maintaining the temporal order in which they were acquired). When producing potential non-consecutive overlapping pairs, this sequence is scanned. Only those frames that have a lower position in the trajectory are taken into account to be associated with the current frame in order to prevent redundancy.

1) Check that the planar motion does not include improper rotations.
2) Check that the 3D motion Euler angles associated to this planar transformation are within a certain range [25].
3) The number of SURF correspondences between the two images must be larger than a certain threshold.
4) The overlapping area between the images must be higher than a threshold.

As a last step, further point correspondences are searched for in the image pairs that passed successfully the above tests. The two images are aligned according to the previously estimated homography. Then, we extract Harris corners [26] in one of the images, and their correspondences are detected through correlation [27] in the other image. If enough correspondences are not found, the pair of images is rejected.

As a result of the processing presented in this section, we produce a set of overlapping pairs with associated point correspondences that will be used in the next section as input for the bundle adjustment.

## IV. Global Alignment

We parameterize the camera trajectory in the most general terms with 6-DOFs (3D position and orientation) using unitary quaternions [28] to prevent singularities in the representation of the camera rotation. Therefore, pose rotation matrices ${}^{C}_{M}R_{j}$ from (6) are converted into a unit quaternions ${}^{C}_{M}q_{j}$. This gives 3 parameters for the position and 4 for the orientation, giving rise to 7 parameters to be estimated for each image frame $j$.

A bundle adjustment [29] procedure will optimize these poses by minimizing a cost function. The cost function is defined as a stack of residuals coming from four different sources. These different sources are described below. The initial guess given to the optimization is the estimated trajectory using navigation data.

### A. Point Matches

For each overlapping pair computed in the previous section, a planar transformation and a set of correspondences was stored. These correspondences are used as input to the bundle adjustment optimization. The bundle adjustment finds a solution for the trajectory parameters by minimizing the norm of a vector of residuals.

These residuals are a function of the input data (point matches and navigation data) and the trajectory parameters. To promote execution speed, we use a subset of correspondences which are obtained by choosing well spread points that accurately represent the homography. Depending on the required planar motion model, 2, 3 or 4 is the minimum number of correspondences that must be used for Euclidean or Similarity, Affine and Projective motion models, respectively [30]. The projective model is used in this work since it is the one able to represent all the degrees of freedom of the trajectory parameterization (3D rotation and 3D translation).

The following equations show how residuals are calculated by using the relative planar homographies between frames $i$ and $j$.

$$ {}^{i}r^{l}_{k} = {}^{i}x^{l}_{k} - {}^{i}_{j}H \cdot {}^{j}\tilde{x}^{l}_{k} \tag{8} $$

$$ {}^{j}r^{l}_{k} = {}^{j}x^{l}_{k} - {}^{j}_{i}H \cdot {}^{i}\tilde{x}^{l}_{k} \tag{9} $$

where, $l = \{1..m\}$ with $m$ being a determined number of correspondences per pair, and $k = \{1..n\}$ with $n$ being a determined number of overlapping image pairs. Planar relative motions ${}^{i}_{j}H$ and ${}^{j}_{i}H$ between image frames $i$ and $j$ are computed by using the absolute planar motions between each frame and the 2D mosaic as it is explained in II.

In (8), ${}^{i}x^{l}_{k}$ is a point detected in the frame $i$ while ${}^{j}\tilde{x}^{l}_{k}$ is its correspondence in the frame $j$ in homogeneous coordinates, these roles are interchanged in (9). This is used to avoid biases in the estimation process resulting from large scale difference between image frames. As an example, for $m = 4$ correspondences per pair, $n = 10$ overlapping images, $4 \times 10 \times 2 \times 2 = 160$ residuals will be added to the stack.

### B. Fiducial Point Readings

For certain surveying and monitoring applications, a number of world points with known $x$ and $y$ coordinates may be available. We refer to these points as fiducial points. Fiducial points can be added as restrictions when some kind of landmark placed on the sea floor is available. Initially, all the 3D fiducial points ${}^{M}X_{0k}$ are projected to the plane $z = 0$ obtaining ${}^{m}x_{0k}$. This is, transforming them from the 3D mosaic frame to the 2D mosaic frame. Then, residuals are computed using the following equation:

$$ {}^{p}r_{k} = {}^{i}x_{k} - {}^{i}_{m}H \cdot {}^{m}\tilde{x_{0k}} \tag{10} $$

where, ${}^{i}x_{k}$ corresponds to the 2D imaged point ${}^{m}\tilde{x_{0k}}$ in the image frame $i$. Each fiducial point introduces two elements in the residual stack, one in $x$ direction and the other in $y$ direction.

### C. LBL Camera Readings

Each available LBL position reading generates 3 residuals (over the $x$, $y$ and $z$ directions), as shown in (11).

$$ {}^{c}r_{k} = {}^{M}T_{0k} - {}^{M}T_{k} \tag{11} $$

where, ${}^{M}T_{0k}$ is the 3D position reading associated to the frame $k$ and ${}^{M}T_{k}$ is the position vector obtained from the pose $k$ that is being optimized.

### D. Angular Camera Readings

Commonly, navigation data provides roll, pitch and heading orientation of the vehicle. These angles are transformed to rotations of the camera with respect to the 3D mosaic frame $\{M\}$. Thus, for each frame, we obtain roll, pitch and yaw as rotations in $x$, $y$ and $z$ axes of the camera respectively. Then, for each camera pose $k$ we compute 3 residuals in the stack, one for each angle.

$$ {}^{a}r_{k} = a_{0k} - a_{k} \tag{12} $$

where, $a_{0k}$ is the vector with the 3 angular readings associated to frame $k$, and $a_{k}$ are the 3 angles associated to the current vehicle pose of frame $k$ that is being optimized.

### E. Cost function

The minimization algorithm minimizes the weighted squared sum of the residual stack described above. This function is shown in (13).

$$\operatorname{argmin}\left(\tau\cdot\sum_{\substack{1\le l\le m\\1\le k\le n}}(\|{}^{i}r_k^{l}\|^2+\|{}^{j}r_k^{l}\|^2)+\lambda\cdot\sum_{k=1}^{o}\|{}^{p}r_k\|^2+\sum_{k=1}^{t}\langle\mu,{}^{c}r_k{}^2\rangle+\sum_{k=1}^{s}\langle\omega,{}^{a}r_k{}^2\rangle\right) \tag{13}$$

where ${}^{i}r_k^{l}$, ${}^{j}r_k^{l}$ and ${}^{p}r_k$ are two-component vectors and ${}^{c}r_k$, ${}^{a}r_k$ are three-component vectors.

Weights for the different measurements are given according to the uncertainty of the sensors. Thus, we use a constant $\tau$ for point and match correspondences and $\lambda$ for fiducial point readings. In the case of camera position and angular readings, as these information may be obtained from different sensors, the optimization algorithm is ready to take into account different weights for each coordinate of the camera position and for each orientation angle. Therefore, $\mu$ is a vector $(\mu_x,\mu_y,\mu_z)$ and $\omega$ is a vector $(\omega_\alpha,\omega_\beta,\omega_\gamma)$.

The minimization of the cost function and the estimation of the trajectory parameters is carried out using Matlab's large-scale methods for non-linear least squares. The optimization algorithm requires the computation of the Jacobian matrix containing the derivatives of all residuals with respect to all trajectory parameters. Fortunately, this Jacobian matrix is very sparse since each residual depends only on a very small number of parameters [29]. As an example, for the optimization results presented in the next section, the percentage of non-zero values is 0.009 %. Furthermore, it has a clearly defined block structure, and the sparsity pattern is constant [31]. These conditions allow for the efficient computation of the Jacobian. In our implementation, analytic expressions were derived and used for computing the blocks of the Jacobian matrix.

## V. Results

The generic framework described in the previous sections was conceived taking into consideration a general setup for image surveys using underwater platforms equipped with position and angular sensors. This framework was applied on a challenging dataset representing a good example of a large-scale deep water survey.

The deep-sea image set used in this paper was acquired by the ARGO II vehicle of the Woods Hole Oceanographic Institution (WHOI) during the LUSTRE'96 cruise over the Lucky Strike hydrothermal vent field. This vent field is located at the Mid-Atlantic Ridge and covers an area over 1.5 square km. The survey pattern comprised large sparse transects, where more than 20,800 images were captured. Unfortunately, the navigation file contains only sparse positioning and angular sensor readings. The survey took approximately 3 days. The outline of the transects is illustrated in Fig. 3.

Position readings were obtained from two different sensors. Coordinates in $x$ and $y$ directions were obtained from a bottom-moored transponder network (LBL). The $z$ coordinate was provided by an acoustic altimeter. LBL measurements were recorded every 60 seconds while the $z$ measurement was provided once per second. Positions are defined with respect to a UTM world coordinate frame $\{W\}$ whose $x$ axis points to the East, the $y$ axis points to the North and the $z$ axis points upwards.

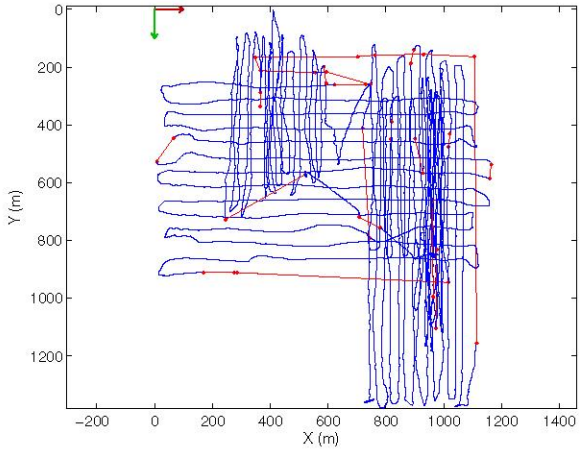



Fig. 3. Estimated trajectory (blue) projected onto the $z = 0$ plane. Only the 20,823 robot poses that have an associated image are depicted. Lines in red show big gaps in the trajectory that identify transects in which either the vehicle was not taking images, or that the survey was stopped and the vehicle was taken out of the water. The top-left corner corresponds to the origin of the 2D mosaic frame $\{m\}$ presented in section II. The scale of the plot is in meters therefore, the surveyed area corresponds to an area around 1.5 $\mathrm{km}^2$.

Orientation readings were provided by two different sensors at a rate of 1 Hz. A fluxgate compass sensed the heading ($\gamma$) as the angular offset from the North ($y$ axis of the $\{W\}$ frame), while an inclinometer measured pitch ($\beta$) and roll ($\alpha$) angles (see Fig. 2) that are deviations of the $z$ axis of the vehicle with respect to the gravity vector in the $y$ and $x$ axis of the vehicle, respectively.

Each entry in the navigation data file is identified by a time stamp with a resolution of 1 second. Therefore, the $x$ and $y$ coordinates of the vehicle were linearly interpolated to yield one reading per second. Only around 5,440 position entries in the navigation file were not interpolated.

One image frame was acquired every 13 seconds and stored using a time stamp identifier. Then, image time stamps were intersected with the time stamps in the navigation file entries to obtain a full pose per image frame. As a result, we obtain a list of 596 images for whose we have a non-interpolated position reading.

Since the LUSTRE'96 dataset does not contain any landmark, $\lambda$ is not used as the second sum in (13) is not considered. The $x$ and $y$ coordinates of the vehicle are given by the LBL system, while $z$ is provided by the altimeter, i.e. $\mu = (\mu_{x_{LBL}}, \mu_{y_{LBL}}, \mu_{z_{Alt}})$. On the other hand, roll and pitch angles were sensed by an inclinometer and heading by a compass, i.e. $\omega = (\omega_{\alpha_{Inc}}, \omega_{\beta_{Inc}}, \omega_{\gamma_{Comp}})$.

A global alignment over the whole LUSTRE'96 dataset have been carried. It required 90 hours in a 1,800 Mhz AMD Opteron™ 265 processor using the Matlab optimization toolbox.

Given the initial mosaic containing 20,823 frames, after overlap detection and filtering step, a connected graph containing 20,226 images as nodes is finally obtained. This produces a non-linear system with $20{,}226 \cdot 7 = 141{,}582$ parameters to be optimized. As constraints for the optimization step, 28,701 overlapping pairs with 4 correspondences per pair were used. This yields $28{,}701 \cdot 4 \cdot 2 \cdot 2 = 459{,}216$ residuals for point and match correspondences. In addition, for each image frame we had 3 angular readings, that adds $20{,}226 \cdot 3 = 60{,}678$ residuals into the stack. Finally, only 596 frames had a non-interpolated position reading. These fixed positions introduce $596 \cdot 3 = 1{,}788$ additional residuals. The final non-linear system had 141,582 parameters and 521,682 residuals to optimize.

Fig. 4 shows a complete view of the mosaic obtained from the navigation data before optimization (left) and after the optimization (right), while Figs. 5 and 6 show the local alignment of a crossing between segments and a consecutive segment, respectively. The mosaic sections were rendered by simple opaque superposition. These areas were obtained from an interactive visualization tool that is available for scientific use [32]. The tool is oriented towards facilitating the browsing of a very large number of registered images, allowing marine scientists to see fine details in the original images while having a geo-referenced global view of the whole survey.

Mosaicing assumes that differences in the 3D relief of the ocean floor are neglectable with respect to the navigation altitude of the vehicle. Under this assumptions, the camera motion does not induce any parallax effects [19]. Obviously, this is not always true in LUSTRE'96 dataset (see Figs. 7 and 8).

## VI. Conclusion and Future Work

This paper described a generic framework for obtaining large area image mosaics acquired by platforms with position and angular sensors, using an off-line bundle adjustment method. It constitutes the initial step in the development of tools intended to be used by marine scientists in benthic mapping applications. A very large image data set was used for testing, representing a good example of a real deep water survey for geology studies.

This approach is based on the assumption of a planar scene and allows building large-area underwater photo-mosaics. The optimization relies in a coherent set of pairwise image correspondences. If false correspondences are introduced, the optimization process cannot achieve the desired solution. For this reason, a three-phase technique is proposed, eliminating those image pairs with inconsistent motion by exploiting SURF matches as a first step, then verifying a set of geometric properties, and finally aligning the images to test correspondences through correlation.

This effective and robust motion estimation allows the system to handle low-overlapping images under arbitrary vehicle motion. It is the first globally-aligned large-scale photo-mosaic of the Lucky Strike vent field at the Mid Atlantic Ridge. The final mosaic contains 20,226 images and has been optimized including slightly more than 28,000 image constraints.

This technique can be readily used with available imagery and associated data to generate optimized seafloor mosaics. These mosaics can then be fully exploited for scientific purposes, including the study of geological, biological, ecological and other features, and their temporal variability if similar mosaics of the same area are available. Existing data from seafloor imagery that remains now unexploited can thus be accessed and studied for scientific purposes.

Ongoing work addresses the use of multiple iterations of image matching and optimization in order to improve the mosaic alignment. After one iteration, the method produces an optimized trajectory of the camera that is transformed to navigation data. This is done by converting back the trajectory to vehicle poses expressed with respect to the world frame $\{W\}$ and extracting $x$, $y$, $z$ position and roll, pitch and heading angles. This new optimized navigation data will be used as starting point for a new iteration of the whole algorithm. This new iteration will find new constraints (i.e. more overlapping image pairs in new crossings) that will lead to a new minimum when running the next bundle adjustment.

## Acknowledgment

This work has been funded in part by MEC [2] under grant CTM2004-04205 and in part by EU under grant MRTN-CT-2004-505026. JF has been funded by MEC under FPI grant BES-2006-12733 and NG has been supported by MEC under the *Juan de la Cierva* program.

The authors would like to thank D. Fornari and S. Humphris from WHOI for providing the LUSTRE'96 dataset, and to Javier Escartín for his helpful comments and suggestions.

## References

[1] R. Garcia, T. Nicosevici, and X. Cufí, "On the way to solve lighting problems in underwater imaging," in *MTS/IEEE OCEANS Conference*, vol. 3, Biloxi, Mississipi, May 2002, pp. 1018–1024.

[2] W. S. Pegau, D. Gray, and J. R. V. Zaneveld, "Absorption and attenuation of visible and near-infrared light in water: dependence on temperature and salinity," *Applied Optics*, vol. 36, pp. 6035–6046, Aug. 1997.

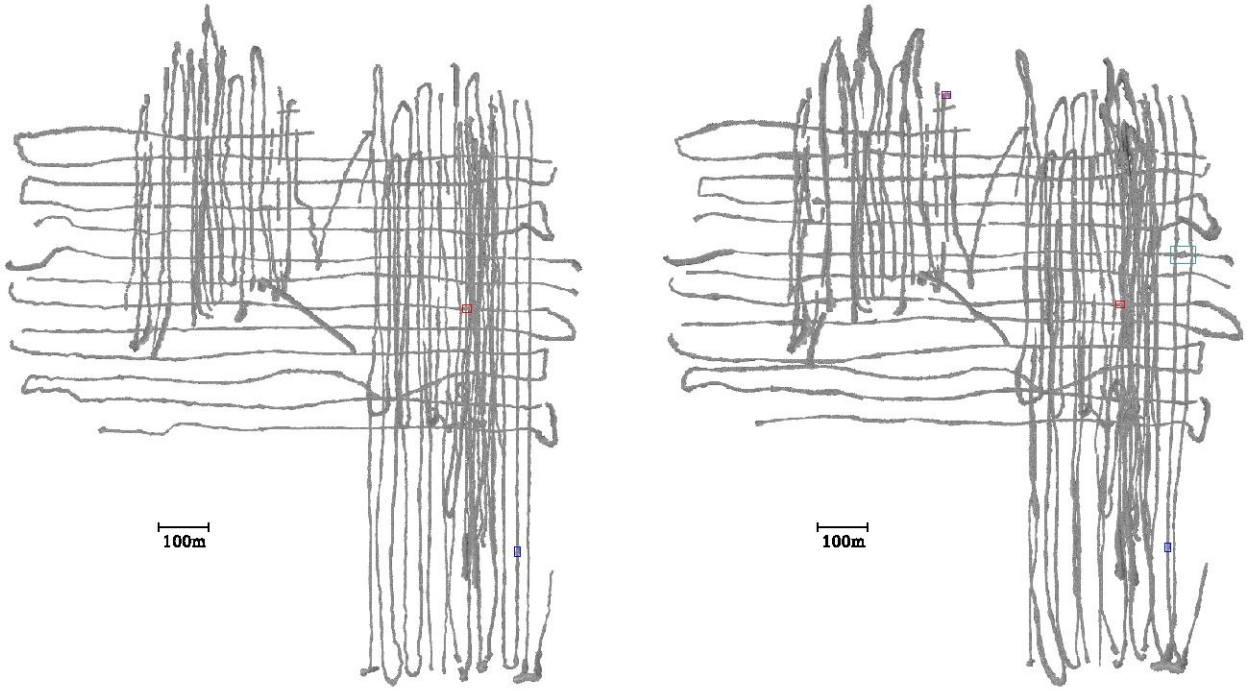
[3] H. Gordon, "Absorption and scattering estimates from irradiance measurements: Monte carlo simulations," *Limnology and Oceanography*, vol. 36, pp. 769–777, 1991.

[4] H. Loisel and D. Stramski, "Estimation of the inherent optical properties of natural waters from irradiance attenuation coefficient and reflectance in the presence of Raman scattering," *Applied Optics*, vol. 39, pp. 3001–3011, 2000.

[5] D. Lirman, N. Gracias, B. Gintert, A. 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," *Environmental Monitoring and Assessment*, vol. 159, pp. 59–73, 2007.

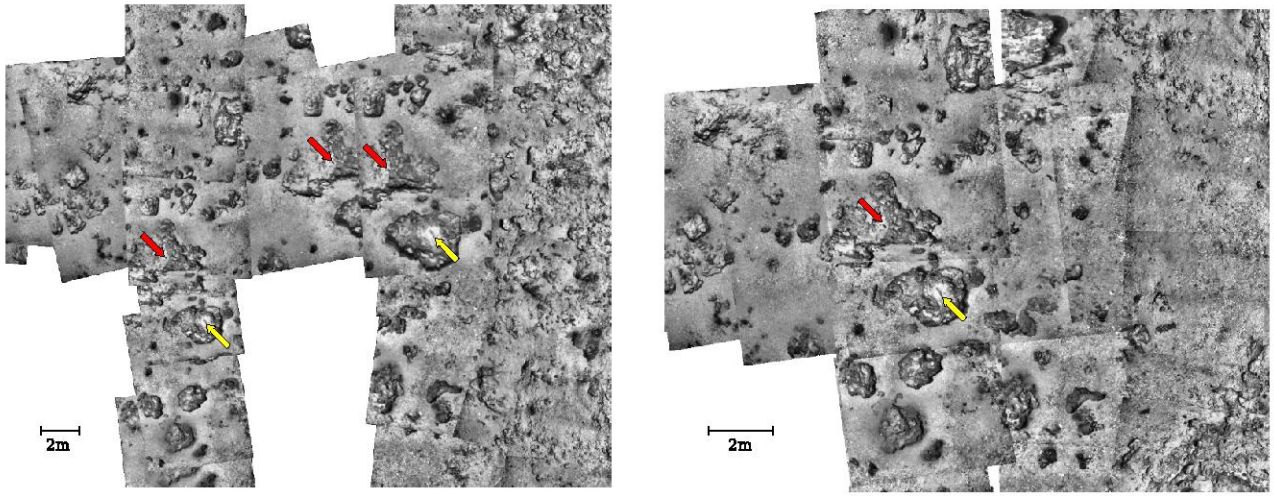
[6] H. Singh, J. Howland, and O. Pizarro, "Advances in large-area photomosaicking underwater," *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 872–886, Jul. 2004.

[2] Spanish Ministry of Education and Science.




(a) Non-optimized (b) Globally Aligned

Fig. 4. Full optical survey comprising 20,226 images. The initial mosaic (a) was obtained using only the navigation data and was refined using bundle adjustment with point and matches between images and LBL and angular reading priors (b). The red and blue rectangles refer to the areas that are enlarged in Figs. 5 and 6, respectively. Parallax effects in the magenta and cyan areas of (b) are shown in Figs. 7 and 8, respectively.




(a) Non-optimized (b) Globally Aligned

Fig. 5. Section of the survey showing a crossing point between several vertical segments and a horizontal one. Inaccuracies in the navigation data result in image misalignments. In image (a), the arrows plotted in the same color point to the same rock. Image (b) shows the resulting alignment after bundle adjustment.

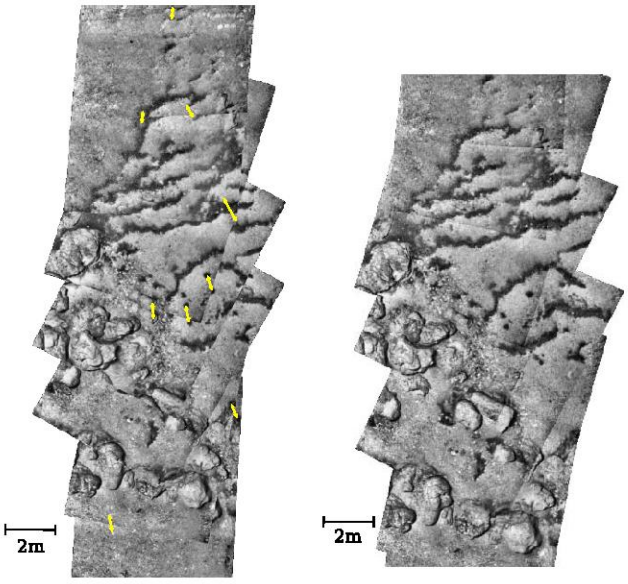
[7] S. Negahdaripour, C. Barufaldi, and A. Khamene, "Integrated system for robust 6-DOF positioning utilizing new closed-form visual motion estimation methods in planar terrains," *IEEE Journal of Oceanic Engineering*, vol. 31, pp. 533–550, 2006.

[8] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS titanic: Conservative covariance estimates for SLAM information filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1223–1242, 2006.

[9] H. Singh, C. Roman, O. Pizarro, R. Eustice, and A. Can, "Towards high-resolution imaging from underwater vehicles," *International Journal of Robotics Research*, vol. 26, no. 1, pp. 55–74, 2007.

[10] P. M. Newman, J. J. Leonard, and R. R. Rikoski, "Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar," in *International Symposium on Robotics Research*, Sienna, Italy, Oct. 2003, pp. 409–420.

[11] D. Ribas, J. Neira, P. Ridao, and J. D. Tardos, "SLAM using an imaging sonar for partially structured environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct.




(a) Non-optimized (b) Globally Aligned

Fig. 6. The low accuracy of the navigation data (a) is noticeable in this mosaic section. The arrows plotted in yellow show misalignments between overlapping images. Image (b) shows the resulting alignment after bundle adjustment.

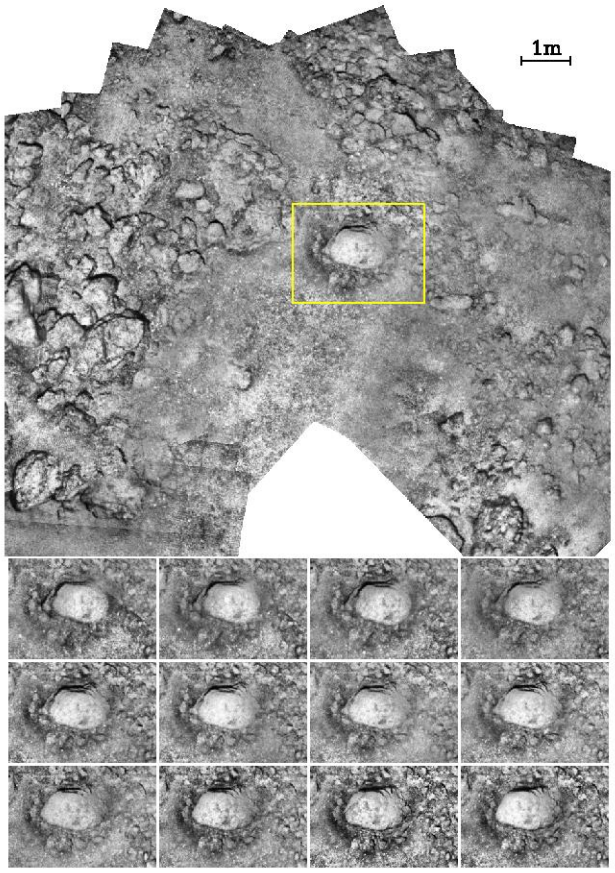



Fig. 7. The top picture enlarges the magenta area in Fig. 4b. The yellow rectangle corresponds to an area which suffers from parallax induced effects. The bottom part of the figure shows 12 cropped tiles of different images that have been registered in this section of the mosaic.

2006.

[12] R. Marks, S. Rock, and M. Lee, "Real-time video mosaicking of the ocean floor," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 229–241, Jul. 1995.

[13] K. Richmond and S. Rock, "An operational real-time large-scale visual mosaicking and navigation system," *Sea Technology*, pp. 10–13, 2007.

[14] O. Pizarro, R. Eustice, and H. Singh, "Relative pose estimation for instrumented, calibrated imaging platforms," in *Digital Image Computing Techniques and Applications*, Sydney, Australia, Dec. 2003, pp. 601–612.

[15] H. Madjidi and S. Negahdaripour, "Global alignment of sensor positions with noisy motion measurements," *IEEE Transactions on Robotics*, vol. 21, pp. 1092–1104, Dec. 2005.

[16] Y. Rzhanov, L. Mayer, S. Beaulieu, T. Shank, S. Soule, and D. Fornari, "Deep-sea geo-referenced video mosaics," in *MTS/IEEE OCEANS Conference*, Boston, USA, Sep. 2006, pp. 2319–2324.

[17] A. Vincent, N. Pessel, M. Borgetto, J. Jouffroy, J. Opderbecke, and V. Rigaud, "Real-time geo-referenced video mosaicking with the matisse system," in *MTS/IEEE OCEANS Conference*, vol. 4, San Diego, USA, Sep. 2003, pp. 2319–2324.

[18] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[20] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Mosaic based navigation for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 3, pp. 609–624, Oct. 2003.

[21] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, Graz, Austria, May 2006, pp. 404–417.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] É. Vincent and R. Laganière, "Detecting planar homographies in an image pair," in *IEEE Symposium on Image and Signal Processing and Analysis*, Pula, Croatia, Jun. 2001, pp. 182–187.

[24] M. Brown and D. G. Lowe, "Recognising panoramas," in *International Conference on Computer Vision*, Nice, France, Oct. 2003, pp. 1218–1225.

[25] B. Triggs, "Autocalibration from planar scenes," in *European Conference on Computer Vision*, Freiburg, Germany, Jun. 1998, pp. 89–105.

[26] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, Manchester, U.K., 1988, pp. 147–151.

[27] R. Garcia, J. Batlle, X. Cufí, and J. Amat, "Positioning an underwater vehicle through image mosaicking," in *IEEE International Conference on Robotics and Automation*, vol. 3, Seoul, Rep. of Korea, May 2001, pp. 2779–2784.

[28] E. Salamin, "Application of quaternions to computation with rotations," Stanford University, Stanford, California, Tech. Rep., 1979.

[29] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – A modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, 2000, pp. 298–375.

[30] N. Gracias and J. Santos-Victor, "Automatic mosaic creation of the ocean floor," in *MTS/IEEE OCEANS Conference*, Nice, France, Sep. 1998, pp. 257–262.

[31] D. P. Capel, *Image Mosaicing and Super-resolution*. Springer Verlag, 2004.

[32] "MosaicViewer," http://eia.udg.es/%7Erafa/mosaicviewer.html.

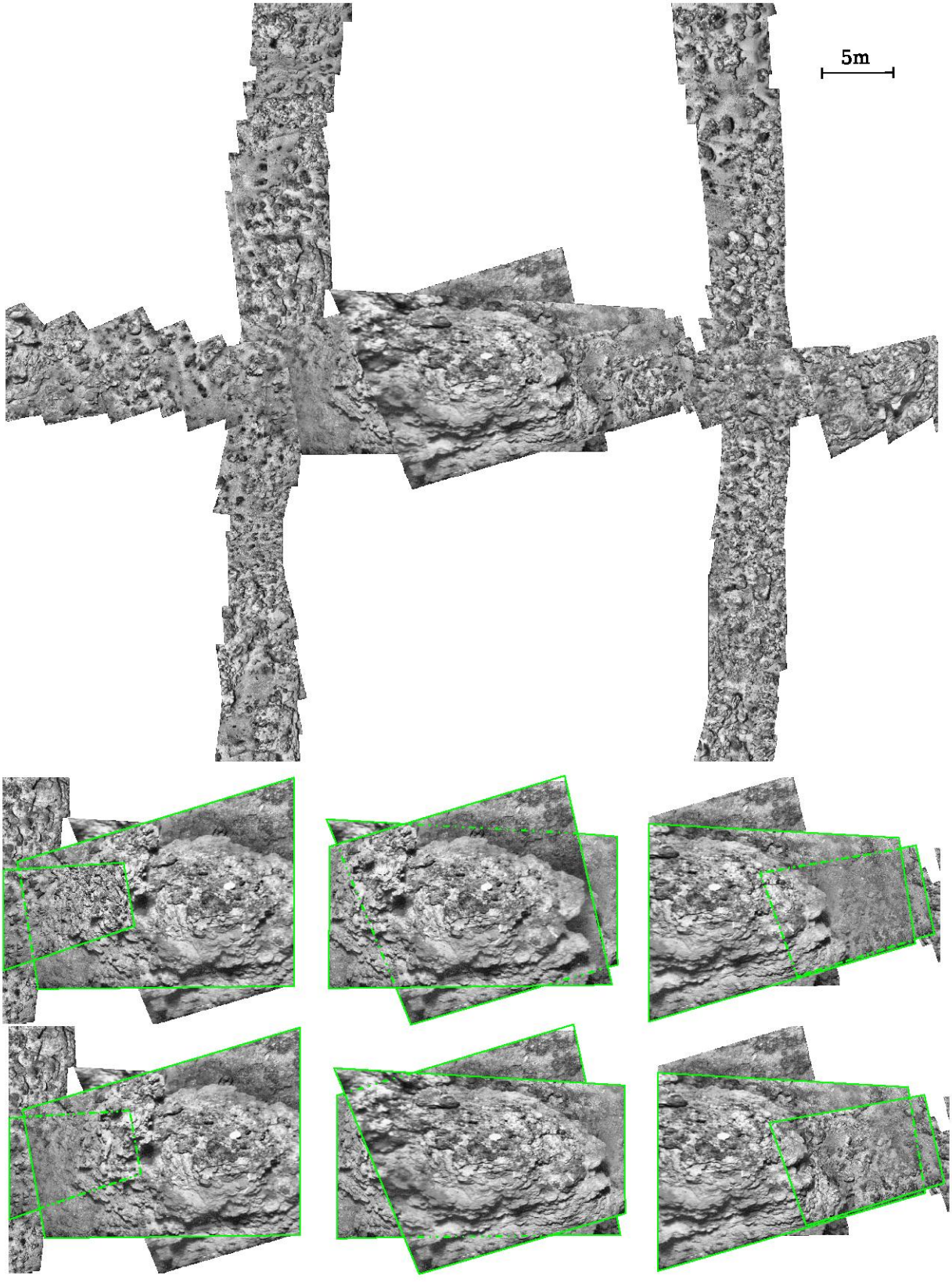



Fig. 8. Zoom of the cyan area from Fig. 4b. This section corresponds to a flat area with a small hill located at the center. The three images in the central row show pairwise image registrations rendering the left-most image on top. The last row renders the right-most on top. Note the strong parallax effects due to big differences in the 3D relief. 2D mosaicing is not able to cope with this problem.

0-933957-35-1 ©2007 MTS