Mosaic Based Navigation for Autonomous Underwater Vehicles

Nuno Gracias, Sjoerd van der Zwaan, Alexandre Bernardino and José Santos-Victor, *Member, IEEE*

***Abstract*— We propose an approach for vision-based navigation of underwater robots that relies on the use video mosaics of the sea bottom as environmental representations for navigation.**

We present a methodology for building high quality video mosaics of the sea bottom, in a fully automatic manner, that ensures global spatial coherency. During navigation, a set of efficient visual routines are used for the fast and accurate localization of the underwater vehicle with respect to the mosaic. These visual routines were developed taking into account the operating requirements of real-time position sensing, error bounding and computational load.

A visual servoing controller, based on the vehicle kinematics, is used to drive the vehicle along a computed trajectory, specified in the mosaic, while maintaining constant altitude. The trajectory towards a goal point is generated online to avoid undefined areas in the mosaic.

We have conducted a large set of sea trials, under realistic operating conditions. This paper demonstrates that, without resorting to additional sensors, visual information can be used to create environment representations of the sea bottom (mosaics) and support long runs of navigation in a robust manner.

***Index Terms*— Underwater computer vision, video mosaics, visual servoing, trajectory reconstruction, uncertainty estimation**

I. Introduction

THIS paper describes a methodology for mosaic-based visual navigation of underwater autonomous vehicles, navigating close to the sea floor. A high-quality video-mosaic is automatically built and used as a representation of the sea-bottom. A visual servoing strategy is adopted to drive the vehicle along a specified trajectory (indicated by waypoints) relative to the mosaic. The control errors are defined by comparing (registering) the instantaneous views acquired by the vehicle and the mosaic. The proposed approach was tested at sea with an underwater vehicle.

The autonomous navigation of underwater vehicles is a growing research and application field. A contributing factor is the increasing need of sub-sea data for activities such has environmental monitoring or geological surveying. Recent interest has been devoted to the development of smart sensors, where the data acquisition and navigation are intertwined. These systems aim at releasing the human operation from low-level requirements, such as the path planning, obstacle avoidance and homing. By providing the platforms with such level of human independence, these systems reduce the operating costs and broaden the potential end-users group. The user main tasks are in the definition of mission primitives to be carried out and higher level mission control.

The underwater environment poses a difficult challenge for precise vehicle positioning. The absence of electromagnetic signal propagation prevents the use of long range beacon networks. Aerial or land robot navigation can rely upon the Global Positioning System to provide real-time updates with errors of just few centimeters, anywhere around the world. The underwater acoustic equivalent is severely limited both in range and accuracy, thus requiring the previous deployment of carefully located beacons, and restricting the vehicle operating range to the area in between. Sonar equipment provides range data and is increasingly being used in topographic matching for navigation, but the resolution is too low for precise, sub–metric navigation. Vision can provide precise positioning if an adequate representation of the environment exists, but is limited to short distances to the floor due to visibility and lighting factors. However, for the mission scenarios where the working locations change often and are restricted to relatively small areas, the use of visual based positioning can be the most appropriate.

The methodology in this paper considers mission scenarios where an autonomous platform is required to map an area of interest and navigate upon it afterwards, as illustrated in Fig. 1.

For building the video mosaic, the quality constraints are very strict, as the mosaic will be the basis for global navigation. Hence, highly accurate image registration methods are needed. The proposed mosaic creation method deals with long image sequences, the automatic inference of the path topology, and the 3–D recovery of the overall geometry. The quality of the mosaic is improved by ensuring global data consistency, using overlapping image regions from loop trajectories or zig-zag scanning patterns.

During navigation, the performance depends heavily upon the ability of the vehicle to localize itself with respect to the previously constructed map. Two important requirement are the bounding of localization errors and real-time availability of position estimates. To address this, two distinct visual routines are run simultaneously: image-to-mosaic registration and inter-image motion estimation.

The first routine, also referred to as *mosaic matching*, is devised for accurate positioning and error bounding. It runs at a low frequency, and uses robust model–based feature matching to register a current image over the mosaic. The outcome of this routine is the 3–D position of the camera in a

Manuscript received ; revised . This work was supported by the EU-Esprit-LTR Proj. 30185 - NARVAL and by the Portuguese Foundation for Science and Technology PRAXIS XXI BD/13772/97.

The authors are with the VisLab, Instituto de Sistemas e Robótica – Instituto Superior Técnico, Av. Rovisco Pais, 1049–001 Lisbon, Portugal. Email – {ngracias, sjoerd, alex, jasv}@isr.ist.utl.pt

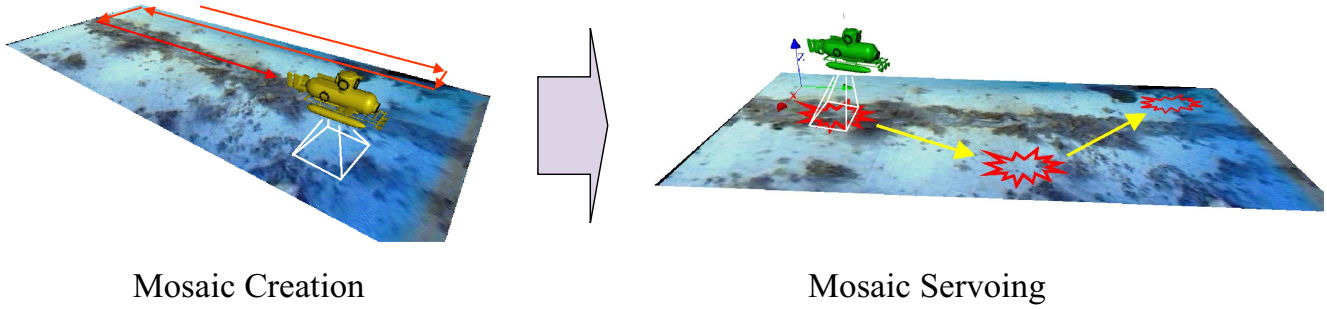

Fig. 1. The operation modes for the proposed mosaic-based navigation system.

world coordinate frame associated with the map. The second routine computes the image motion between camera images in real–time. It provides updates over the last successful mosaic matching, but is prone to accumulate errors if not combined with the mosaic matching.

Depending on the use, the localization information can either be expressed directly in terms of an image frame (e.g. pixel location on the mosaic) or converted to an explicit position and orientation of the vehicle in 3D.

To avoid driving the vehicle over the areas where the mosaic matching may be too difficult (such as near the borders of the region covered by the mosaic), a trajectory generation module was implemented. This module provides a set of waypoints between the current and final location that simultaneously searches for a short travel path while keeping away from the mosaic borders.

A final control module consists in translating all these data into control errors and design the controllers that drive the vehicle propellers.

This paper builds upon our previous work on mosaic creation, pose estimation and station keeping. In [1], [2] the basic process of sequential image processing is detailed for the creation of underwater mosaics, along with the pose estimation and error propagation. The present paper extends this approach to loop trajectories and distant superpositions, which are essential for the creation of spatially coherent mosaics over large areas.

In [3] a fast template–based pose estimation method and a vision–based controller are derived. These methods are specially suited for underwater station keeping [4]. We use the same controller structure, but extend it to navigate to distant points with respect to the starting position. Furthermore, a trajectory generation procedure is also implemented. A key point is the extension and integration of several methods for mosaic creation, pose estimation and visual servoing that complement each other.

Part of this work done in the context of the European Project NARVAL [5]. The main scientific goal was the design and implementation of reliable navigation systems for mobile robots in unstructured environments, without resorting to global positioning methods. The algorithms and results described in this paper, where large mosaics are created and used for posterior navigation, constitute a major achievement towards this goal.

A. Related Work and Contributions

This paper relates to the work of a number of authors. The most important differences are now highlighted.

1) Mosaic Construction with Global Registration: The application of mosaicing techniques for underwater imagery is a topic of increasing research interest, not only as a visualization tool for covering large areas [6], [7], [8], [9], but also as a spatial representation for underwater robotics [10], [11], [12]. Comparative results on vehicle positioning and mosaicing for long image sequences are reported in [13], where calibrated testing of the algorithm presented in this paper is included.

Considering the topic of global registration, several approaches have been proposed using topology inference of neighboring frames [14], [15], and restricted parameterizations for the projection matrices [16]. Recent methods allow the fast computation of globally consistent linear strips mosaics [17], and use Kalman Filtering for closing the trajectory loops [18].

The main differences of our approach with all of the above are twofold: (1) the parameterization of the homographies with complete and meaningful 3D pose parameters and, (2) the inclusion of the unknown single world plane constraint.

2) Mosaics for Navigation: One of the early references to the idea of using mosaics as visual maps is the work of Zheng [19], where panoramic representations were applied to route recognition and outdoor navigation. However the visual representations do not preserve geometric characteristics nor correspond to visually correct mosaics. This constitutes a drawback as the representation is not fit for human perception, which is important for mission definition. Recently, Kelly [20] has addressed the feasibility and implementation issues of using large mosaics for robot guidance, predicting a large impact of these techniques on industrial environments. In this case, the problem is simplified by assuming that the image plane is parallel to the mosaiced areas and the motion of the vehicles is restricted to the ground plane.

Xu [21] investigated the use of seafloor mosaics, constructed using temporal image gradients, in the context of concurrent mapping and localization, for real-time applications. Albeit careful compensation for systematic errors, eventual loops in the camera path are not taken into account nor used for compensating for the accumulated error, which prevents the use in covering large areas. Huster [22] described a navigation interface using live–updated mosaics, and illustrated the advantages of using it as a visual representation for human

operation. However, since the mosaic is not used in the navigation control loop, there is no guaranty the vehicle is driven to the desired position.

One of the works more closely related to ours is [11], in the sense it combines spatially consistent mosaic with underwater ROV navigation. However, in their approach, the navigation system requires additional sensors to provide heading, pitch and yaw information, whereas our work relies solely upon vision to provide information for all the relevant degrees of freedom. The integration of several different sensors benefits for the robustness of the overall navigation system, provided that realistic estimates exist for the uncertainty associated with each sensor. However, it is of scientific relevance to know of far can underwater vision systems go when used alone, and have ways of computing the uncertainty of the pose. Our paper is directed towards this goal.

Regarding navigation, our approach differs from the concurrent mapping and localization approaches (CM&L, SLAM) in the sense the map is totally created prior to its use. Some authors have successfully implemented (and extended) CM&L for the underwater navigation with mosaics [21], [11], which has the advantage of using the mosaic while it is being constructed, but leads to less accurate mosaics due to the real–time constraints.

3) Contributions: Our paper presents the first results on using large mosaic for servoing at sea, which validates the servoing approach. Previous experiments on visual trajectory following using mosaics at sea were restricted proof-of-concepts, using very small mosaics and few meter runs [21], [11].

Our work contributes to the field of visual underwater navigation in several ways:

- The mosaic creation is approached in a *fully automated and integrated* way where *global spatial consistency* is imposed by estimating the image neighboring topology.
- *All the degrees of freedom* arising from the mosaic geometry are taken into account and parameterized as geometrically meaningful entities – pose parameters and world plane description. As a result, we retrieve the vehicle trajectory over time, together with the associated uncertainty.
- Different techniques of inter-image motion and image-to-mosaic matching were devised and used in a combined manner, together with robust estimation methods, providing the necessary degree of accuracy and robustness.
- A simple and effective visual servoing control scheme is used to drive the vehicle. The error signals are defined exclusively from image measurements.
- The appropriateness of the approach is demonstrated by successful experimental testing in the challenging, real–world conditions of the underwater environment (at sea).

B. Paper Organization

Section II reviews some useful entities and methods related with the mosaic geometry. Section III details the algorithm used for creating a navigation map through image mosaicing, along with illustrative results. Section IV describes the mosaic navigation, namely the algorithms implemented for the vehicle localization, the image based control law and selected results from the mosaic servoing experiments. Section V draws the conclusions and establish directions for future work.

II. Visual Geometry and Motion Estimation

This section reviews some important geometry entities and estimation methods related to the mosaicing process.

A. Camera model

The adopted camera model is the standard pin-hole model, which performs a linear projective mapping of the 3D world into the image frame. The camera is assumed to be calibrated with known intrinsic parameter K matrix [23], [24]. In this work, the radial distortion was estimated and used for the off-line creation of displacement maps. During the image acquisition, these maps are used as look-up tables for the on-line correction of the distortion.

B. Homographies

We assume that the working area of the sea bottom can be approximated by a plane. Two views of the same 3–D plane are related by a homography [25] (also referred to as a planar transformation) which is represented by a 3×3 matrix defined up to a scale factor. A homography H performs a point–to–point mapping between the homogeneous coordinates of the image points $\mathbf{x}'$ and $\mathbf{x}$, such that $\mathbf{x}' = H\mathbf{x}$. It has, at most, eight degrees of freedom which are illustrated in Fig. 2. The estimation of H requires at least four pairs of corresponding points. In the case of more than four correspondences, it can be estimated by least-squares [26].

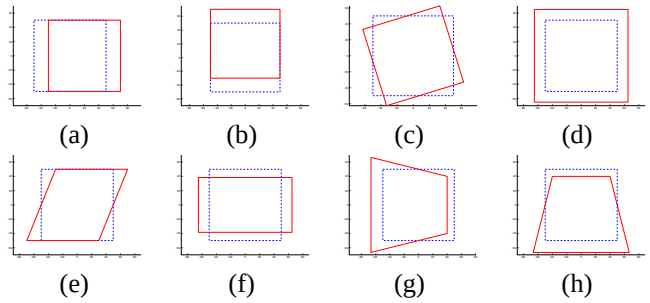

Fig. 2. Degrees of freedom of the planar projective transformation on images: (a) horizontal translation, (b) vertical translation, (c) rotation, (d) scaling, (e) shear, (f) aspect ratio, (g) projective distortion along the horizontal image axis, (h) projective distortion along the vertical image axis

C. Visual Motion Estimation

The starting point for the creation of mosaics is the estimation of image motion between consecutive frames of a video sequence.

For each image I_k, a set of point features is extracted using the Harris corner detector [27] and matched over the following image I_{k+1}, through a correlation-based procedure. The correlation is performed using a fast implementation [28] of the Lucas-Kanade point tracker [29]. A robust estimation

technique is used to filter out matching outliers, and estimate the homography $H_{k,k+1}$ that relates the coordinate frames of I_k and I_{k+1}. A variant of random sampling LMedS, detailed in [1], is used.

A different approach to the computation of image motion was also investigated for station keeping applications. In this approach, the tracking of an image region is performed by combining optic flow information with template matching, which relies on a set of pre–computed motion models to describe the image deformations. The robustness of the method is increased by adapting the set of used models to the most commonly observed camera motions [3], [30].

D. 3–D Pose Estimation

Given an homography, it is possible to obtain the relative 3–D motion of the camera up to a scale factor. A homography matrix H_{21} is decomposed [31] as

$$H_{21} = K\left(R_{21} + t\frac{n_1^T}{d_1}\right)K^{-1} \tag{1}$$

where R_{21} and t are, respectively, the 3×3 rotation matrix and the 3×1 translation vector relating the 3-D camera frames. The world plane (that induces the homography) is accounted for by the unitary vector n_1, containing the outward plane normal expressed in the camera 1 coordinates, and the distance d_1 of the plane to the first camera center.

The problem of recovering the motion parameters from a homography for an intrinsically calibrated camera is discussed in-depth by Faugeras [31]. In the most general case there are eight different sets of solutions. However, only two are valid for a non-transparent world plane, which are found using the SVD of $M_{21} = K^{-1}H_{21}K$ [32].

Most often it is important to estimate not only the vehicle location but also the associated uncertainty, in a 3–D world referential. The pose uncertainty allows for monitoring the performance of the localization algorithm, and provides information for sensor fusion in the case of using more than one positioning modality. We will now outline how the uncertainty can be propagated from the point matches to the pose parameters, using a first order approximation.

Let $\Theta = \begin{bmatrix} \alpha & \beta & \gamma & {}^W_C\mathbf{t}_x & {}^W_C\mathbf{t}_y & {}^W_C\mathbf{t}_z \end{bmatrix}^T$ be the 6-vector containing the camera pose in the form of 3 camera rotation angles and the location of the camera centre in world coordinates. The 3-D rigid transformation that relates points in the world and camera frames is given by

$$\begin{bmatrix} {}^C\mathbf{x} \\ {}^C\mathbf{y} \\ {}^C\mathbf{z} \end{bmatrix} = {}^CR_W\left(\begin{bmatrix} {}^W\mathbf{x} \\ {}^W\mathbf{y} \\ {}^W\mathbf{z} \end{bmatrix} - \begin{bmatrix} {}^W_C\mathbf{t}_x \\ {}^W_C\mathbf{t}_y \\ {}^W_C\mathbf{t}_z \end{bmatrix}\right). \tag{2}$$

The rotation matrix ${}^CR_W$ is defined by X-Y-Z fixed angle convention[33],

$${}^CR_W = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \tag{3}$$

where $s(.)$ and $c(.)$ represent the sine and cosine functions of the rotation angles.

Without loss of generality, we assume that all the world points belong to the plane defined by ${}^W\mathbf{z} = 0$. A image-to-world homography has the form :

$$\Psi(K,\Theta) = K\cdot{}^CR_W\cdot\begin{bmatrix} 1 & 0 & -{}^W_C\mathbf{t}_x \\ 0 & 1 & -{}^W_C\mathbf{t}_y \\ 0 & 0 & -{}^W_C\mathbf{t}_z \end{bmatrix}. \tag{4}$$

If a set of matches exists between image point projections and their world coordinates, then the camera pose can be estimated [2], by minimizing the following error function

$$\begin{aligned} F(X,\Theta) = \textstyle\sum_{i=1}^N \quad & [d^2\left(\mathbf{x}_i, \Psi(K,\Theta)\cdot\mathbf{x}_i'\right) \\ & +d^2\left(\mathbf{x}_i', \Psi^{-1}(K,\Theta)\cdot\mathbf{x}_i\right)] \end{aligned} \tag{5}$$

where $\mathbf{x}_i$ and $\mathbf{x}_i'$ denote corresponding image and world–plane points, and $d(\cdot,\cdot)$ is the Euclidean distance.

In this paper, the uncertainty propagation follows the method described by Haralick in [34] for propagating the covariance matrix through any kind of linear or non-linear calculation. The method assumes that a scalar function, $F(X,\Theta)$, is defined which is minimized by the noisy estimate $\widehat{\Theta}$, and noisy data $\widehat{X}$, and that the calculation can be well approximated by a first order Taylor series expansion for the level of noise involved.

An estimator for the 6×6 covariance matrix $\Sigma_{\Delta\Theta}$ of the noise in $\widehat{\Theta}$, is given by

$$\Sigma_{\Delta\Theta} = J\cdot\Sigma_{\Delta X}\cdot J^T \tag{6}$$

where $\Sigma_{\Delta X}$ is the covariance matrix of the data, and

$$J = \left[\frac{\partial^2 F}{\partial\Theta^2}\left(\widehat{X},\widehat{\Theta}\right)\right]^{-1}\left[\frac{\partial^2 F}{\partial X\partial\Theta}\left(\widehat{X},\widehat{\Theta}\right)\right]^T. \tag{7}$$

E. Uncertainty propagation from the pose to the mosaic

If the camera pose and associated uncertainty are known, then we can estimate the location where the camera optical axis intersects the mosaic, and its uncertainty. This is helpful in defining search areas for the initial matching over the mosaic map. The intersection of the camera optical axis with the world plane is given by Eq. (2) with the additional constraints of ${}^C\mathbf{x} = {}^C\mathbf{y} = {}^W\mathbf{z} = 0$. This system is easily solvable for the intersection coordinates,

$$\begin{bmatrix} {}^W\mathbf{x} \\ {}^W\mathbf{y} \end{bmatrix} = {}^W_C\mathbf{t}_z\cdot\begin{bmatrix} \frac{s\beta}{c\beta c\gamma} \\ -\frac{s\gamma}{c\gamma} \end{bmatrix} + \begin{bmatrix} {}^W_C\mathbf{t}_x \\ {}^W_C\mathbf{t}_y \end{bmatrix}. \tag{8}$$

For small levels of noise, $\Upsilon(\Theta) = \begin{bmatrix} {}^W\mathbf{x} & {}^W\mathbf{y} \end{bmatrix}^T$ can be approximated by its first-order Taylor expansion. The associated covariance matrix $\Sigma_{\Delta\Upsilon}$ is approximated by $\Sigma_{\Delta\Upsilon} = J\cdot\Sigma_{\Delta\Theta}\cdot J^T$, where J is the partial derivatives matrix of $\Upsilon(\Theta)$,

$$J = \begin{bmatrix} 0 & \frac{{}^W_C\mathbf{t}_z}{(c\beta)^2 c\gamma} & \frac{{}^W_C\mathbf{t}_z\, s\beta s\gamma}{c\beta(c\gamma)^2} & 1 & 0 & \frac{s\beta}{c\beta c\gamma} \\ 0 & 0 & -\frac{{}^W_C\mathbf{t}_z}{(c\gamma)^2} & 0 & 1 & -\frac{s\gamma}{c\gamma} \end{bmatrix}. \tag{9}$$

III. MOSAIC MAP CREATION

The mosaic map creation method comprises four major stages, illustrated in Fig. 3. Firstly, the image motion is computed in a sequential manner to infer the approximate topology[1] of the camera movement. Secondly, this topology is refined by searching for non–consecutive matches on areas where the path winds up on itself. Next, a global minimization is carried out, using the most general 6-degree of freedom motion model and the point matches between all the images. Finally, the images are blended to create a fronto–parallel mosaic image, and a 3–D world referential is associated with it.

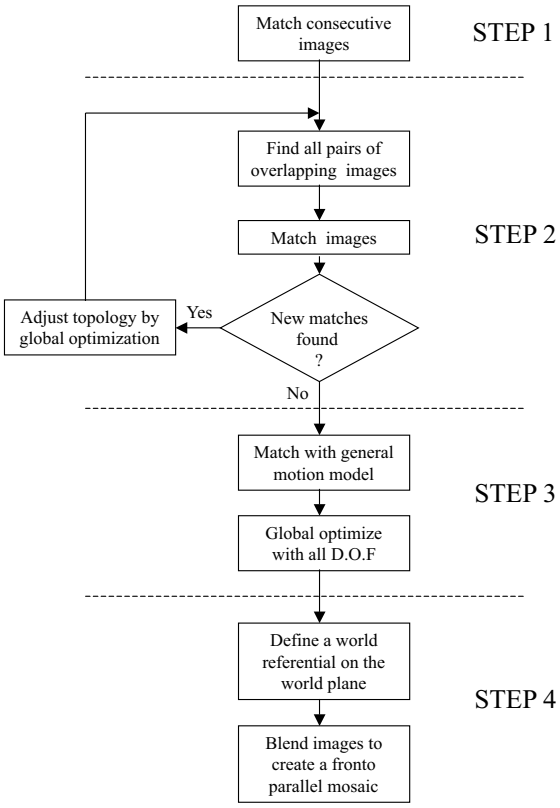

Fig. 3. Flow–chart for the complete mosaic creation algorithm.

A. Step 1 - Initial Motion Estimation

The first part of the algorithm consists on the sequential estimation of inter-frame homographies, as described in Section II-C.

In order to speed up the initial matching process, the computed homography for the current pair of images is used to restrict the correlation search over the next pair. If, after the random sampling LMedS, the image matching is not successful then the process is repeated with larger correlation areas.

In underwater vision, the image acquisition rate is usually high when compared to the camera motion, resulting in large overlap between consecutive frames. To avoid this, a frame discarding procedure was used during acquisition, thus reducing the memory and processing requirements for the next stages. A minimum of overlap of 60 % is imposed for the selected frames. This threshold was chosen based on the results of preliminary matching trials.

This step is performed on-line during the mosaic image acquisition. As a by product, it allows for the real-time creation of simple mosaics, without global constraints. This proves to be very useful for the maneuvering of the vehicle during the acquisition, as it provides visual information on the approximate trajectory of the vehicle.

B. Step 2 - Iterative Topology Estimation

After the initial motion estimation step, possible overlap between non-consecutive images can be predicted, and used to search for new image matches.

In this stage, the topology is estimated by performing iterative steps of image matching and global optimization. The image matching part is conducted over overlapping frames, and is similar to what was described above. If new matches are found, then the topology is re-estimated by means of a global optimization procedure. This procedure uses a reduced representation for the camera motion, based on 3 parameters (2D translation and rotation), that implicitly assumes the camera is facing the ground at a constant distance. The reason for a simpler motion model for the first two stages of the algorithm, has to do with the effectiveness of the topology inference. The most general 8–parameter homographies can cope with general perspective distortion, but has more degrees of freedom than usually required. Consequently, small errors in the initial inter–frame motion estimation tend to quickly accumulate, and make it impossible to infer the neighboring relations among non-consecutive frames.

The cost function to be minimized is the sum of distances between each correctly matched point and its corresponding point after being projected onto the same image frame, *i.e.*,

$$F(X,\Theta) = \sum_{i,j}^{} \sum_{n=1}^{N_{i,j}} \begin{array}{l} [d^2\left(x_n^i, H(\Theta_i,\Theta_j)\cdot x_n^j\right) \\ \quad + \; d^2\left(x_n^j, H^{-1}(\Theta_i,\Theta_j)\cdot x_n^i\right)] \end{array} \tag{10}$$

where $N_{i,j}$ is the number of correct matches between frame i and j, and $H(\Theta_i,\Theta_j)$ is the homography constructed using the motion parameter vectors Θ_i and Θ_j. These vectors contain the 3 parameters that relate the frames i and j with the reference frame (first frame). The minimization is carried out using a non-linear least squares algorithm [35], and the cycle of matching and topology refinement is executed until no new image pairs can be matched.

To speed-up the optimization procedure (and, thus, the motion refinement cycle time), a sub-mosaic aggregation scheme was implemented. Under this scheme the complete sequence is initially divided into sets of consecutive images to form small rigid sub-mosaics. Inside each sub-mosaic the homographies are considered static and only the inter-mosaic homographies are taken into account in the optimization algorithm. This

[1]The topology refers to the complete set of inter–image homographies that allows for the registration of all images on a common image frame, thus implicitly containing the camera path information in image coordinates.

reduced parameter scheme significantly improves the speed of evaluating the cost function and does not affect the capability of inferring the appropriate trajectory topology.

C. Step 3 - High Accuracy Global Registration

The main objective of the final stage of the algorithm is attaining a highly accurate registration. A more general parameterization for the homographies is therefore required, capable of modelling the warping effects caused by wave-induced general camera rotation and changes on the distances to the sea floor. Therefore, a parameterization is devised to take explicitly into account all the 6 degrees of freedom of the camera pose.

The estimation of the homographies using a general model does not impose, *per se*, the existence of a single world plane from which the homographies are induced. This condition is imposed by augmenting the overall estimation problem with additional parameters that describe the position and orientation of the world plane. Such additional parameters are included on the parameterization of the homographies.

An important advantage of the devised parameterization is that it allows for the full 3–D camera trajectory and world plane to be recovered during the process. Although this knowledge is not explicitly used for the navigation methods of this paper, it is nonetheless valuable in the context of an integration mission.

1) General parameterization: One of the camera frames (usually the first) is chosen as the origin for the 3-D referential, where the optical axis is coincident with the referential Z-axis. The world plane is parameterized with respect to this frame by 2 angular values that define its normal. As the trajectory and plane reconstruction can only be attained up to an overall scale factor, this ambiguity is removed by setting the plane distance to 1 metric unit[2], measured along the Z-axis.

Let Θ_i and Θ_j be the pose 6-vectors containing 3 rotation angles and 3 translations with respect to the reference 3-D frame of the first camera. Let $n\left(\Theta_p\right)$ be a 3-vector containing the normal to the world–plane (also in the 3-D reference frame), which is parameterized by the 2-vector Θ_p of angles. The homography relating frames i and j with the reference image frame is given by Eq. (1):

$$\begin{aligned} H_{i,1} &= K \cdot \left[R\left(\Theta_i\right) + t\left(\Theta_i\right) \cdot n^T\left(\Theta_p\right)\right] \cdot K^{-1} \\ H_{j,1} &= K \cdot \left[R\left(\Theta_j\right) + t\left(\Theta_j\right) \cdot n^T\left(\Theta_p\right)\right] \cdot K^{-1} \end{aligned} \tag{11}$$

where $R\left(\Theta_i\right)$ and $R\left(\Theta_j\right)$ are rotation matrices, $t^T\left(\Theta_i\right)$ and $t^T\left(\Theta_j\right)$ are the translation components, as defined in Section II-D. The homography relating frames i and j is given by

$$\begin{aligned} H_{i,j} &= H_{i,1} \cdot H_{j,1}^{-1} \\ &= K \cdot \left[R\left(\Theta_i\right) + t\left(\Theta_i\right) \cdot n^T\left(\Theta_p\right)\right] \cdot \\ &\quad \left[R\left(\Theta_j\right) + t\left(\Theta_j\right) \cdot n^T\left(\Theta_p\right)\right]^{-1} \cdot K^{-1} \end{aligned} \tag{12}$$

[2]If additional information is available on the real distance to the sea floor (for example, from an altimeter), then it can be straightforwardly used here.

2) Cost Function: The cost function is similar to the one previously used in the iterative motion refinement, where the distances between matched points are measure in their respective image frames, and summed over all pair of correctly matched images, *i.e.*,

$$\begin{aligned} F(X,\Theta) = \sum_{i,j} \sum_{n=1}^{N_{i,j}} & \left[d^2\left(x_n^i, H_{i,j} \cdot x_n^j\right)\right. \\ & \left. + d^2\left(x_n^j, H_{i,j}^{-1} \cdot x_n^i\right)\right] \end{aligned} \tag{13}$$

For a set of M images, the total number of parameters to be estimated is $(M-1) \times 6 + 2$.

The initialization values for the complete parameter set are computed using Eq. (1). As there are two valid solutions for the decomposition of the homographies relating each frame with the reference frame, the solutions are chosen such that the variance of the world plane normals is minimized. The considered world plane normal is the average of the selected set. As before, the cost function is minimized using non-linear least squares.

D. Step 4 - Mosaic 3-D Referential and Image Blending

For the navigation, we are interested in establishing an Euclidean 3-D world reference associated with the mosaic. As its location is purely arbitrary, the origin is set at the intersection of the optical axis of the first image with the plane of the mosaic. The orientation is such that the mosaic plane has null $\overrightarrow{z}$ coordinate, and the $\overrightarrow{x}$ axis is parallel to the first camera frame $\overrightarrow{x}$ axis. If the information about overall scale is available from a sensor such as an altimeter, it is also used here. As the orientation of the world plane is explicitly taken into account and estimated, it is straightforward to compute the planar projective transformation that yields a fronto-parallel view of the mosaic[3].

The final operation consists of blending the images, i.e., choosing the representative pixels to compose the mosaic image, taken from the spatially registered images. A common method is using the last contributing image. However, considering the use for navigation, an alternative method is used. The mosaic is created by choosing the contributing points which were located the closest to the center of their frames. In underwater applications, it compares favorably to other commonly used rendering methods such as the average or the median, since it better preserves the textures and minimizes the effects of unmodelled lens distortion, which is larger at the image borders.

E. Mosaic Construction Results

The results reported in this paper were obtained from experiments conducted using a custom modified commercially available Phantom 500SP ROV. The ROV is illustrated in Fig. 4 and among other sensors, is equipped with a pan and tilt camera. The controllable degrees of freedom are defined

[3]The most appropriate projection for the visaul map is the fronto-parallel, since it minimizes the perspective image distortions in the image-to-mosaic matching for vehicles where the camera is pointing downwards.

by the geometric arrangement of the thrusters. The forward–backward force and a differential torque are applied by two horizontally placed thrusters while an upward–downward force is applied by a vertical thruster. This arrangement creates non-holonomic motion constraints. The ROV is wired to a remote processing unit by a 150 meter umbilical cable. Video signals are sent up to the ground surface for processing.

Fig. 4. Computer controlled Phantom ROV with the on–board camera. The camera housing is visible in the lower right, attached to the crash frame.

A set of experiments using the remotely operated vehicle were conducted at sea. The ROV was deployed from a pier, and operated within the umbilical cord range of 100 meters. For this range the water depth varied between 2 and 7 meters. Although the working area was fairly flat, a large percentage of area was covered with waving algae. Some sandy pits proved adequate for the map construction and navigation tests. Several successful experiments were conducted from which the most representative are now presented.

An illustrative example of the mosaic creation process is given in Fig. 5. The image sequence was acquired in shallow waters of about 2 meters depth, while the vehicle was manually driven around a squared shaped rock. During the acquisition, the inter-frame motion estimation was performed on-line, which allowed for the selection of 98 images based on a 60% superposition criteria. This resulted in the upper–left mosaic, where the effects of error accumulation are visible near the image top in the form of a repeated white stone. After 4 steps of topology estimation, 285 distinct pairs of non-consecutive images (combined from the selected image set) where successfully matched and the final topology of the upper-right mosaic was obtained. Next, the global optimization was carried out using full 3-D pose parameters for all cameras and world plane description. The outcome of this step allows for the creation of a fronto-parallel view of the mosaic, which can be used as the navigation map.

The high quality of the final mosaic is testified by the fact that visual features lying on the predominant ground plane, such as the small algae covered rocks, are not disrupted along the visible boundaries of the contributing images.

For the servoing tests, the mosaic of Fig. 6 was used, created from 46 selected images. The original image sequence was acquired over a sandy area delimited by algae. An on-board sensor measured the distance from the sea floor to the position where the first image of the sequence was captured. The measured value of 4.29 meters was then used to set the overall mosaic map scale. The mosaic covers approximately 64 square meters, from which 26 correspond to sand. Each pixel on the mosaic corresponds to a sea floor area of about 2×2 centimeters. The rectangular region that contains the mosaic area measures 10.8×9.5 meters. The mosaicing process was able to successfully cope with image contents that clearly departs from the assumed planar and static conditions. This is visible in the large percentage of the mosaic area used by moving algae.

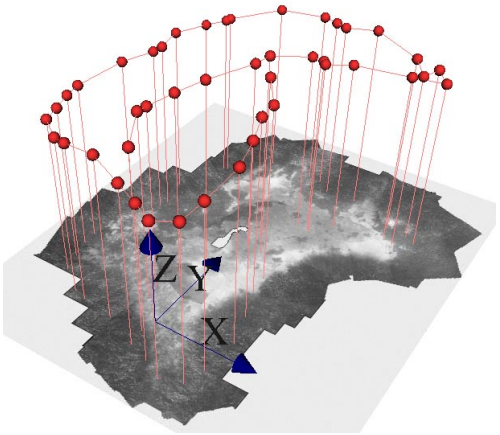

Fig. 6. Perspective view of the mosaic used for the underwater navigation tests, and original camera path. The dots mark the 3–D position of the camera centres for the set of 46 images selected to create the mosaic. The world referencial is represented by the 3 axes. Vertical lines were added to ease the perception of the 3–D trajectory.

IV. Mosaic Navigation

Having described the methods for creating mosaics as visual maps, we will now address the problem of navigation. The mosaic based navigation comprises 3 distinct modules: localization, guidance and control. An overall block illustration of the main modules is shown in Fig. 7.

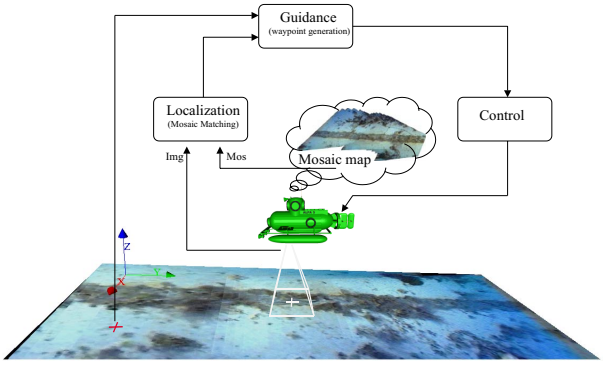

Fig. 7. Overall visual servoing control scheme.

A. Localization on the Mosaic

The first step of mosaic localization consists of finding the initial match between the current camera image and the corresponding area on the mosaic. To restrict the search area, a coarse estimate is required of the vehicle 3D position and orientation, with respect to the mosaic world referential.

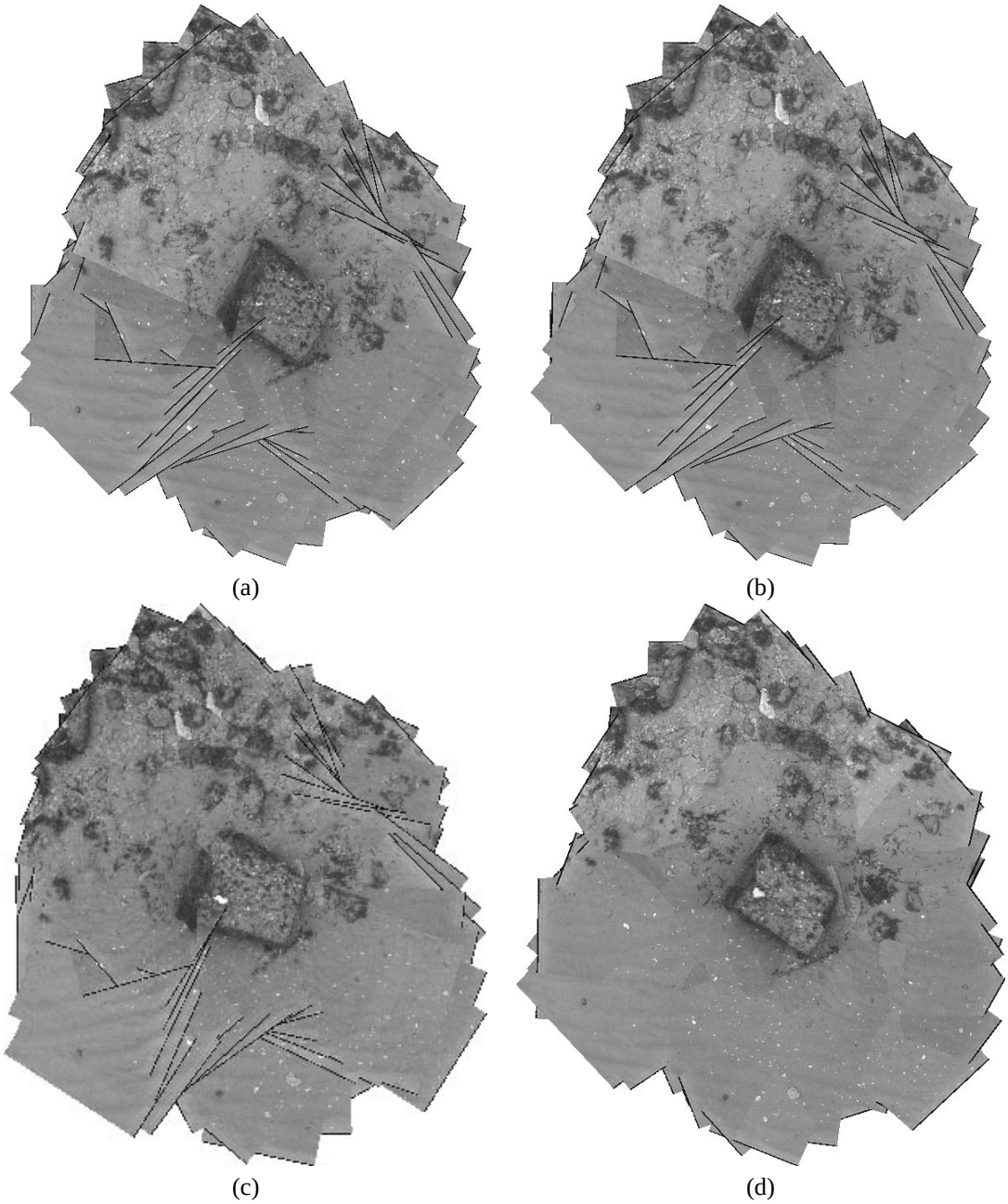

Fig. 5. Mosaic creation example with intermediate step outcome – Consecutive image motion estimation (a), topology estimation and and non-consecutive image matching (b), high accuracy global registration (c) and final fronto-parallel view of the mosaic after global optimization (d). The first three mosaics were rendered using the pixels from the last contributing image, while the last was created with the contribution from the image whose pixels were closer to the frame center.

This will typically be provided by some other modality of autonomous navigation in which a coarse global position estimate is maintained, such as beacon-based navigation or surface GPS reading. This estimate is used for searching for point matches. If the associated uncertainty is available, then it is used to bound the search area on the mosaic, as mentioned in Section II-E.

If the search for point matches is not successful on the first attempt, then a spiral search pattern is used for the subsequent tries. This pattern defines new areas in the mosaic where the search will be performed, thus no motion of vehicle is required. To find the appropriate distance between search areas, a set of experiments was conducted using typical underwater images and mosaics.

For the experimental part of this paper, no external modality of positioning was available to provide the required initial pose estimate. Therefore, this pose was computed from a very coarse matching of 3 points, that were manually provided.

1) On–line tracking: The on–line tracking comprises two complementary processes which run in parallel, at very distinct rates.

Absolute localization – The current image is matched directly over the mosaic, to have an absolute position estimate. This procedure is similar to the initial match, in

the sense it uses the current position estimate to restrict the search area, and the spiral search pattern in case of the initial matching failure.

Incremental tracking – This process estimates the incremental vehicle motion by matching pairs of images from the incoming video stream. The motion model used is a four-parameter homography that accounts for 3-D translation and rotation over the vertical axis. The success of the image matching is assessed by the percentage of correctly matched points found. In the case of unreliable measures, occurring when the number of selected matches is close to the minimum required for the homography computation, the resulting homography is discarded and replaced by the last reliable one.

The complementary nature is illustrated by the fact that the two processes address different requirements of the position estimation needed for control and navigation: real-time operation and bounded errors. The mosaic matching is a time-consuming task (as it might not be sucessful on the first attempt) but provides an accurate position measurement. Conversely, the image-to-image tracking is a much faster process, but due to its incremental nature tends to accumulate small errors over time, eventually rendering the estimate useless for our control purposes, if used by itself. It is also worth noting that this scheme is well fit for multiprocessor platforms, as the two processes can be run separately.

The contributions from the two processes are combined by simply cascading the image-to-image tracking homographies over the last successful image-to-mosaic matching. A typical position estimation update rate of 7 Hz is attained, on a dual–processor machine.

The considered image motion model for the incremental tracking is the four-parameter homography. This model assumes fronto-parallelism of the image plane with respect to the scene, and is more restrictive than the most general six d.o.f. model. However, the four-parameter model was found to be the best trade-off between (1) accurate motion representation capability and (2) insensitivity to estimation noise, which causes accumulated error build-up.

B. Trajectory Generation

A simple trajectory generation algorithm was used for the navigation experiments. The purpose of generating trajectories is to make the vehicle avoid the map areas where the mosaic matching is likely to fail. Examples of such are the areas of non-static algae, the mosaic borders or regions that were not imaged during the mosaic acquisition phase. In this paper, only the distance to the mosaic borders was considered, but the method can be used to avoid any region defined *a priori* in the map.

The first step consists of the creation of a cost map associated with navigating over every elementary region of the map image, where the regions to be avoided have higher costs than the rest. The cost map is created by using the distance transform [36], and contains positive values that decrease with the distance to the border of the valid region. Outside the valid region, the cost is set to a sufficiently large positive number.

Given the current and desired vehicle positions on the mosaic, we want to find the path that minimizes the sum of costs. This is formulated as a minimal path problem, where a path is defined as an ordered set of weighted locations. To solve it efficiently, we use Dijkstra's algorithm [37], whose complexity is $O\left(m^2\right)$ where m is the number of pixels in the cost image. Fig. 8 presents an example of the generation of trajectories using this method.

The cost map is created *off-line*, after the mosaic creation phase. During operation, a new trajectory is generated *on-line* each time an end-point is specified. For the purpose of avoiding the mosaic edges, a relatively small number of trajectory waypoints is required. Therefore, the size of the cost image can be reduced so that the computation of the trajectory does not compromise the on-line nature of the mosaic servoing.

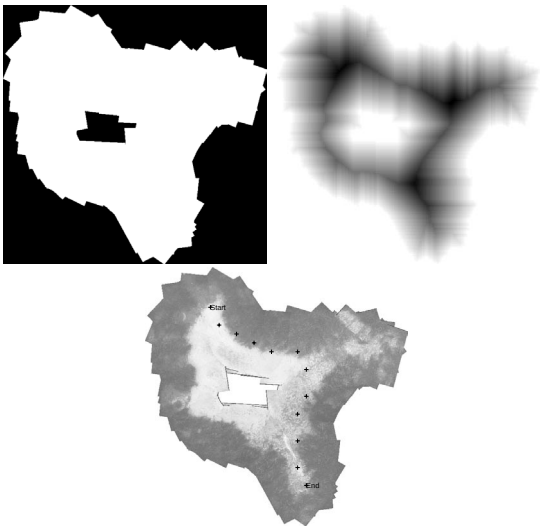

Fig. 8. Trajectory generation example – Valid mosaic region in white (upper left), cost image (upper right) and mosaic with superimposed trajectory (lower image).

C. Control

Motivated by the under–actuated nature of the vehicle, a *decoupled control design* is adopted, which controls the motion of robot in the horizontal plane, and maintains a constant altitude in the vertical plane. The controller is design within the framework of visual servoing strategies [38], and makes use of direct measurements on the image coordinates as opposed to the use of 3–D pose information. Even though 3–D information is available on-line at little added computational cost, the image-based approach attained higher performance on preliminary tests.

The implemented controllers were developed for visual station keeping and docking applications [3], [39], and are based on the approaches of Espiau *et al.* [40] and Malis *et al.* [41]. Details on the model identification and low-level control of the platform can be found in [42].

1) Servoing over the Mosaic: Servoing to a goal position on the mosaic is defined as the regulation to zero of an image error function $\mathbf{e}\left(\mathbf{s}\right) = \mathbf{s} - \mathbf{s}_d$, where $\mathbf{s}$ is the image feature

parameter vector and $\mathbf{s}_d$ the desired value. The image center of the current camera image is used as a feature, whose desired position is at some (distant) docking point on the mosaic, as illustrated in Fig. 9. The image error function is then given by $\mathbf{e} = [x_c, y_c]^T - [x_d, y_d]^T$ where (x_c, y_c) is the projection of the current image center onto the mosaic and (x_d, y_d) represents the docking point on the mosaic. Note that the projection of the center of the current image onto the mosaic is calculated based on the mosaic localization procedure, which provides the control system with the current image-to-mosaic homography. The objective of the controller is to drive the projected image center towards the docking position, rejecting external disturbances.

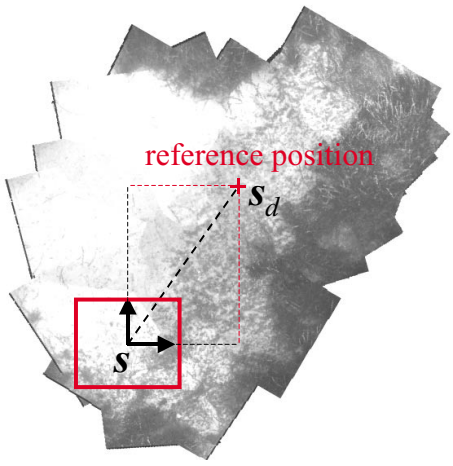

Fig. 9. Definition of error measures on the mosaic. The current image frame is represented by the frame rectangle and the reference is marked by the cross.

Changes in the image features, $\dot{\mathbf{s}}$, can be kinematically related to changes in the relative camera pose, by the *image Jacobian* matrix L [38], [40]:

$$\dot{\mathbf{s}} = \mathrm{L} \cdot \mathbf{v}_{cam} \tag{14}$$

where $\mathbf{v}_{cam}$ is the 6×1 camera velocity screw. The image Jacobian for the image center is given by :

$$\mathrm{L}(\mathbf{s}, Z) = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x_c}{Z} & x_c y_c & -(1+x_c^2) & y_c \\ 0 & -\frac{1}{Z} & \frac{y_c}{Z} & (1+y_c^2) & -x_c y_c & -x_c \end{bmatrix}. \tag{15}$$

This Jacobian depends both on the image point coordinates and their depth, Z. An exponential decrease of the error function is obtained by imposing $\dot{\mathbf{e}} = -\lambda \cdot \mathbf{e}$, with λ some positive constant. Using Eq.(15), we can then solve for the camera motion that guarantees this convergence:

$$\mathbf{v}^*_{cam} = -\lambda \cdot \mathrm{L}(\mathbf{s}, Z)^+ \cdot (\mathbf{s} - \mathbf{s}_d) \tag{16}$$

where $\mathbf{v}^*_{cam}$ is the resolved camera velocity that comprises the control objective and L^+ is the pseudo-inverse of the image Jacobian.

Since the robot's control input is defined, in general, in the vehicle reference frame, it is useful to relate the vehicle velocities to camera velocities. This relationship is given by the vehicle-to-camera Jacobian, designated by:

$$\mathbf{v}_{cam} = \mathrm{J}_{r2c} \cdot \bar{\mathbf{v}}_{robot} \tag{17}$$

where $\bar{\mathbf{v}}_{robot}$ contains the controllable velocity components of the vehicle velocity screw and J_{r2c} is the robot-to-camera Jacobian relationship. This Jacobian is a function of the camera position and orientation in the vehicle reference frame, $\mathrm{J}_{r2c} = f\left({}^{rov}R_{cam}, P_{cam}\right)$. If the camera position and orientation are available beforehand, the Jacobian can be easily computed from transforming linear and angular velocity components between the frames. It is now possible to reformulate the control objective in terms of desired vehicle velocity components, such that the image center is driven towards the docking point over the mosaic. Also, this Jacobian allows to take the vehicle motion constraints into account by considering only the vehicle controllable degrees of freedom, thus resulting into physically executable trajectories.

Substituting (17) into (14), we obtain an expression that relates the image motion to the vehicle velocity:

$$\dot{\mathbf{s}} = \mathrm{L} \cdot \mathrm{J}_{r2c} \bar{\mathbf{v}}_{robot} \ . \tag{18}$$

With this expression, we can solve for the vehicle velocity in the horizontal plane, necessary to guarantee the convergence of the image error function:

$$\bar{\mathbf{v}}^*_{robot} = -\lambda \cdot \left(\mathrm{L}(\mathbf{s}, Z) \cdot \mathrm{J}_{r2c}\right)^+ (\mathbf{s} - \mathbf{s}_d) \ . \tag{19}$$

Figure 10 illustrates the structure of the visual servoing controller utilized in this paper. The term B^{-1} is part of the dynamic model of the vehicle's thrusters, and allows the computation of the necessary forces and motor torques that correspond to the required (steady state) velocities.

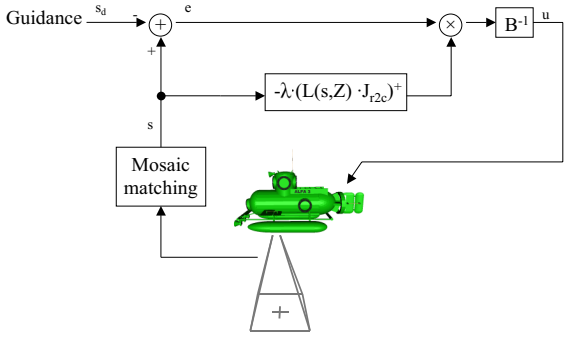

Fig. 10. Control block diagram.

2) Auto–altitude control: The controller for the vertical plane aims at maintaining the camera at a fixed altitude during navigation. This is achieved by regulating to zero the difference in scale between the current image–to–mosaic homography and a reference value.

The image scaling induced by an affine transform homography Ψ can be computed from the determinant of the upper left 2×2 submatrix [43]:

$$s = \sqrt{|\Psi_{2\times 2}|} \ . \tag{20}$$

This does not hold for general projective homographies (as the scale changes along the image), due to the projective distortion. However, this is a suitable approximation if the camera plane is approximately parallel to the ground floor.

The scaling factor s from the current image-to-mosaic homography is compared to a reference scaling, to generate the control error

$$e = s - s_d$$

where the reference s_d is taken from the initial image–to–mosaic homography. The control signal are generated using a PID control action in the form

$$u_{robot} = -(K_p \cdot e + K_d \cdot \dot{e} + K_i \cdot \int e \; dt) \qquad (21)$$

where the gains were manually tuned.

D. Mosaic Servoing Results

An illustrative underwater servoing experiment is presented in Fig. 11 where a top-view of the ROV trajectory and references are plotted. The ROV completed several loop trajectories and travelled for 159 meters, during a 7 minute run. The references where manually specified through a simple user-interface, where the operator was required to click over the desired end position. A more detailed view of part of the run is given in Fig. 12, corresponding to 42 seconds.

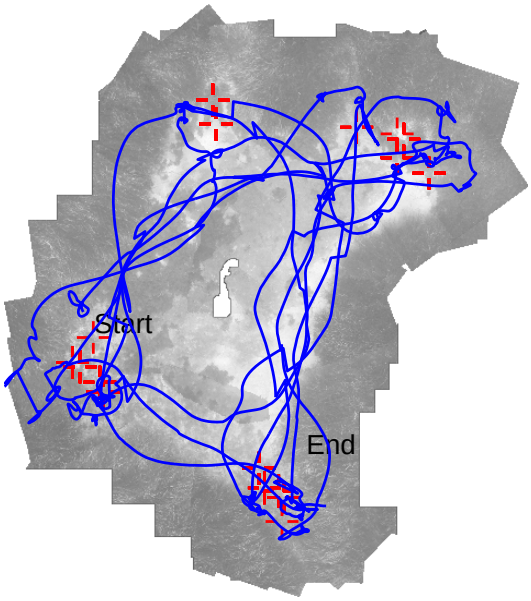

Fig. 11. Underwater mosaic servoing experiment. This plot shows a top–view of the ROV trajectory for the complete run with the reference positions marked with crosses. The ROV trajectory was recovered for the on–line image–to–mosaic matching with updates from the image–to–image tracking, and is marked with the full line.

During the servoing experiments all the images that where matched over the mosaic, where also recorded on disk. As an off-line processing step, these images were re-matched over the mosaic, and the point correspondences where used to estimate both the full 6 d.o.f. pose and the associated uncertainty [26]. For this computation, the following assumptions were considered:

- the only source of uncertainty was the limited accuracy on the point matching,
- point matches were affected with Gaussian noise, uncorrelated over the two coordinates,

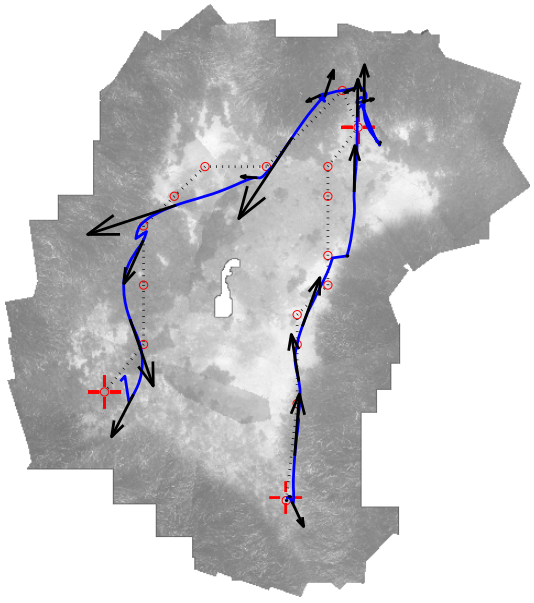

Fig. 12. Trajectory detail is presented comprising two endpoints. The generated path connecting the endpoints is marked by the dashed line. In order to allow the sense of speed, a set of arrows is superimposed. The arrows are drawn every 2 seconds and sized proportionally to the platform velocity.

- the standard deviation was 0.5 pixels for all coordinates. This value was experimentally measured from the residuals of the homography estimation [2].

The ellipsoidal uncertainty volumes associated with the translational part of the pose parameters, are represented in Fig. 13. From the relatively flat, horizontally-levelled ellipsoids, it can be seen that the uncertainty on the camera position is larger along the $\overrightarrow{x}$ and $\overrightarrow{y}$ axes than along the $\overrightarrow{z}$.

As stated above, in Section IV-A, the on-line tracking comprises two complementary processes of position estimation, running simultaneously but at distinct rates. The mosaic matching was triggered in fixed intervals of 5 seconds, typically requiring 3 seconds to be complete if it was successful on the first attempt. The image–to–image tracking ran permanently over consecutive pairs of incoming images, and was used to update the current position estimate at approximately 7Hz. The image processing and servoing was run on a dual–processor 800MHz computer.

During sea trials, the set of images used by the image–to–image tracker were recorded on disk for latter processing. This allowed for the off-line matching of the whole sequence over the mosaic, using the same algorithms as during the on-line mosaic matching. The trajectory was recreated using the 4 d.o.f. fronto-parallel parameterization for the pose and compare it to the on-line estimates, which combined the incremental image–to–image tracking estimate with the last available mosaic matching.

Figure 14 plots the horizontal metric distance between the camera centres for the on-line and off-line estimates, during a selected period of 60 seconds. The duty cycle of the mosaic matching is represented as a square wave, where the rising edge corresponds to the acquisition of a new image to be matched over the mosaic, and the falling edge corresponds

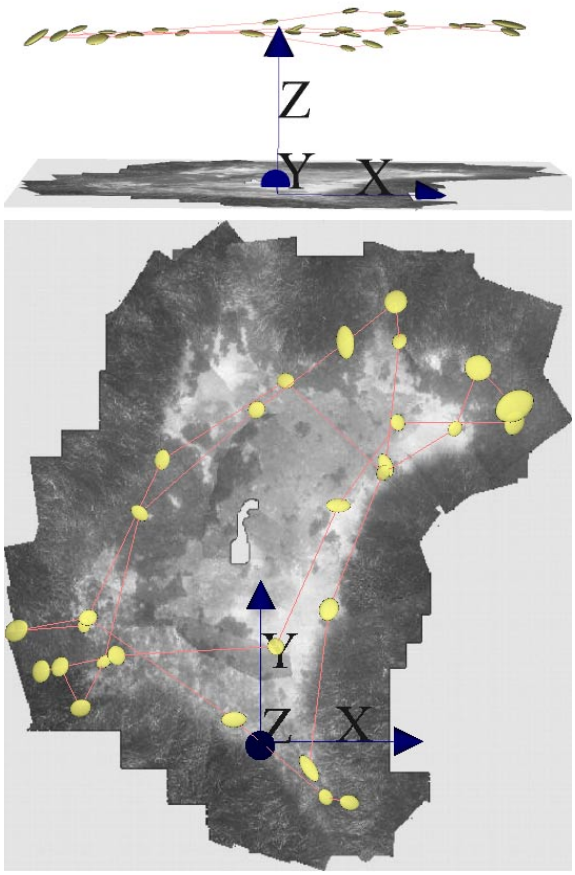

Fig. 13. Mosaic servoing trajectory reconstruction – The two views show the camera positions associated with the images that were directly matched over the mosaic during the servoing run. The ellipsoids mark the estimated camera centers and convey the uncertainty assotiated with the translation part of the pose. The ellipsoid dimensions are set for a 50% probability. However, for clarity reasons, the ellipsoid axes sizes were enlarged by a factor of four, and only 143 seconds of the run are represented.

to the instant when the mosaic matching information becomes available. The error does not fall to zero during the mosaic updates. This is due to the fact that the mosaic-based estimate is only available some time after the corresponding image was acquired, thus allowing for the error to grow in between.

This plot illustrates the need and importance of the periodic mosaic matching, which is apparent from the fast error build-up between mosaic matches, and in its fall once the matching is successful. This approach also presents the advantage of allowing the monitoring of the accumulated error during the on-line run, which can be directly measured immediately after a successful mosaic match. Although not implemented, the magnitude of the accumulated error can be used to adjust the frequency of the image–to–mosaic matching, thus adapting to cases where the image–to–image tracking performance changes.

The reason for not doing exclusively image–to–mosaic matching during the navigation, is the processing time involved. As stated above, the image–to–mosaic matching requires typically 3 seconds, if the matching is successful in the first try. Otherwise it can take much longer. Conversely the image–to–image can run at 7Hz.

The difference in the processing times has to do with the dissimilarity between the on-line camera images and the corresponding areas in the mosaic. This is mainly due to the non-planarity and non-rigidity of the scene, and to illumination changes over time. It is also due to implementation issues, since during the image–to–mosaic matching we match a much larger number of correspondences and apply feature warping prior to the correlation.

Even if the intervals between mosaic matches were reduced and a smaller number of correspondences were searched for, it would be difficult to achieve a position update frequency suitable for the visual servoing. However, being a computational issue, this trade-off between precision and availability is much dependant on the computing resources available.

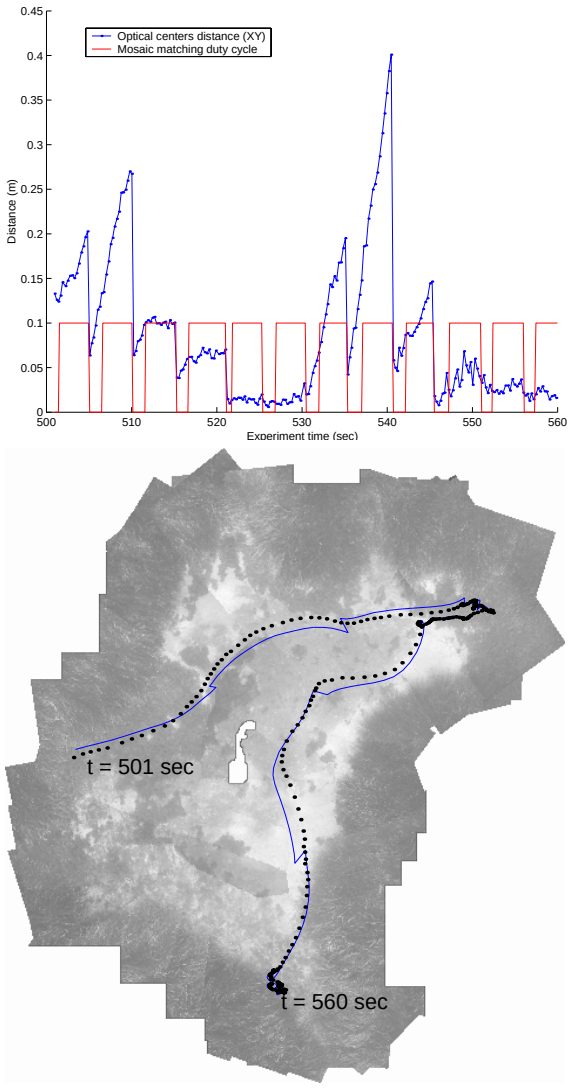

Fig. 14. Diference between the online position estimate, using mosaic matching with inter–image tracking updates, and the offline estimate, obtained by maching all the images over the mosaic. The upper figure shows the horizontal (XY) distance, where the online mosaic matching instants are marked with small circles. On the lower figure the online trajectory is plotted as solid line, while the offline is marked in dash.

The results on mosaic based servoing show the feasibility of using vision as a single positioning modality for relatively large distances, and extended periods of time. The devised

methods allow for positioning with bounded errors through periodic mosaic matching. Also, the uncertainty from the point matches is taken into account, thus allowing for the prediction of the pose estimation accuracy.

V. Conclusions

We have presented a methodology for mosaic based visual-servoing for underwater vehicles for missions where the vehicle is asked to approach a distant point.

Video mosaics are proposed as visual representations of the sea bottom. We have developed a general and flexible approach for building high-quality video mosaics, able to cope with a very general camera motion and guaranteeing the overall spatial coherency of the mosaic. As a by-product, the vehicle trajectory can be recovered together with the associated uncertainty. A set of visual routines were proposed for localizing the vehicle with respect to the mosaic. The matching schemes differ on requirements in terms of execution speed and accuracy. Particular care was taken to ensure the degree of robustness necessary for processing underwater imagery.

A path planning method was applied to ensure that the vehicle avoids navigating near the borders of the valid regions of the mosaic, thus increasing the chances of correctly positioning itself.

A visual control scheme, based on image measurements, was proposed to drive the vehicle. It attained good overall performance for the trajectory following, given the underactuated nature of the test bed, and the the fact that no dynamic model of the vehicle motion was used.

The methodology was tested at sea, under realistic and adverse conditions. It showed that it was possible to navigate autonomously over the previously acquired mosaics for large periods of time, without the use of any additional sensory information.

We believe that the use of large video mosaics as environment representations and as a support for navigation allows for the development of a rich set of navigation modes that can significantly extend the operational autonomy of mobile vehicles acting in unstructured environments.

Several open problems and improvements will be addressed in the future. When building the video mosaic, we plan to develop a strategy to ensure the complete coverage of the region of interest, during the image acquisition process, avoiding area gaps and guaranteeing the existence of sufficient area overlap between swaths for the topology estimation.

References

[1] N. Gracias and J. Santos-Victor, "Underwater video mosaics as visual navigation maps," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, July 2000.

[2] ——, "Trajectory reconstruction with uncertainty estimation using mosaic registration," *Robotics and Autonomous Systems*, vol. 35, pp. 163–177, July 2001.

[3] S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Visual station keeping for floating robots in unstructured environments," *Robotics and Autonomous Systems*, vol. 39, no. 3–4, pp. 145–155, June 2002.

[4] S. van der Zwaan and J. Santos-Victor, "Real–time vision–based station keeping for underwater robots," in *Proc. of the IEEE Oceans 2001 Conference*, Honolulu, Hawaii, U.S.A., November 2001.

[5] "NARVAL – Navigation of Autonomous Robots via Active Environmental Perception, Esprit–LTR Project 30185," 1998–2002. [Online]. Available: vislab.isr.ist.utl.pt/NARVAL/index.htm

[6] R. Eustice, H. Singh, and J. Howland, "Image registration underwater for fluid flow measurements and photomosaicking," in *Proc. of the Oceans 2000 Conference*, Providence, Rhode Island, USA, September 2000.

[7] E. Trucco, A. Doull, F. Odone, A. Fusiello, and D. M. Lane, "Dynamic video mosaics and augmented reality for subsea inspection and monitoring," in *Proc. of the Oceanology International Conference 2000*, Brighton, England, March 2000.

[8] E. Trucco, Y. Petillot, I. T. Ruiz, C. Plakas, and D. M. Lane, "Feature tracking in video and sonar subsea sequences with applications," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 92–122, July 2000.

[9] H. Singh, J. Howland, L. Whitcomb, and D. Yoerger, "Quantitative mosaicking of underwater imagery," in *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[10] S. Negahdaripour, X. Xu, A. Khamene, and Z. Awan, "3D motion and depth estimation from sea-floor images for mosaic-based positioning, station keeping and navigation of ROVs/AUVs and high resolution sea–floor mapping," in *Proc. IEEE/OES Workshop on AUV Navigation*, Cambridge, MA, USA, August 1998.

[11] S. Fleischer, "Bounded–error vision–based navigation of autonomous underwater vehicles," Ph.D. dissertation, Stanford University, California, USA, May 2000.

[12] X. Xu and S. Negahdaripour, "Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles," in *Proc. International Conference on Robotics and Automation (ICRA2001)*, Seoul, Korea, May 2001, pp. 2759–2766.

[13] S. Negahdaripour and P. Firoozfam, "Positioning and image mosaicing of long image sequences; Comparison of selected methods," in *Proc. of the IEEE Oceans 2001 Conference*, Honolulu, Hawai, USA, November 2001.

[14] H. Sawhney, S. Hsu, and R. Kumar, "Robust video mosaicing through topology inference and local to global alignment," in *Proc. European Conference on Computer Vision*. Springer-Verlag, June 1998.

[15] E. Kang, I. Cohen, and G. Medioni, "A graph–based global registration for 2D mosaics," in *Proc. of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000.

[16] K. Duffin and W. Barrett, "Globally optimal image mosaics," in *Graphics Interface*, 1998, pp. 217–222.

[17] R. Unnikrishnan and A. Kelly, "Mosaicing large cyclic environments for visual navigation in autonomous vehicles," in *Proc. International Conference on Robotics and Automation (ICRA2002)*, Washington DC, USA, May 2002, pp. 4299–4306.

[18] R. Garcia, J. Puig, P. Ridao, and X. Cufi, "Augmented state Kalman filtering for AUV navigation," in *Proc. International Conference on Robotics and Automation (ICRA2002)*, Washington DC, USA, May 2002, pp. 4010–4015.

[19] J. Zheng and S. Tsuji, "Panoramic representation for route recognition by a mobile robot," *International Journal of Computer Vision*, vol. 9, no. 1, pp. 55–76, October 1992.

[20] A. Kelly, "Mobile robot localization from large scale appearance mosaics," *International Journal of Robotics Research (IJRR)*, vol. 19, no. 11, 2000.

[21] X. Xu, "Vision–based ROV system," Ph.D. dissertation, University of Miami, Coral Gables, Miami, May 2000.

[22] A. Huster, S. Fleischer, and S. Rock, "Demonstration of a vision–based dead–reckoning system for navigation of an underwater vehicle," in *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[23] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 323–344, 1987.

[24] J. Heikkilä and O. Silvén, "A four–step camera calibration procedure with implicit image correction," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[25] O. Faugeras, *Three Dimensional Computer Vision*. MIT Press, 1993.

[26] N. Gracias, "Mosaic–based Visual Navigation for Autonomous Underwater Vehicles," Ph.D. dissertation, Instituto Superior Técnico, Lisbon, Portugal, June 2003. [Online]. Available: vislab.isr.ist.utl.pt

[27] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings Alvey Conference*, Manchester, UK, August 1988, pp. 189–192.

[28] *Open Source Computer Vision Library*, Intel Corporation, 2001. [Online]. Available: www.intel.com/research/mrl/research/opencv/index.htm

[29] B. Lucas and T. Kanade, "Optical navigation by the method of differences," in *Proc. of the Ninth International Joint Conference on Artificial Intelligence*, vol. 2, Los Angeles, California, USA, August 1985, pp. 981–984.
[30] A. Bernardino, J. Santos-Victor, and G. Sandini, "Foveated active tracking with redundant 2D motion parameters," *Robotics and Autonomous Systems*, vol. 39, no. 3–4, June 2002.
[31] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, September 1988.
[32] B. Triggs, "Autocalibration from planar scenes," in *Proc. of the European Conference on Computer Vision*, Freiburg, Germany, June 1998, pp. 89–105.
[33] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.
[34] R. Haralick, "Propagating covariance in computer vision," in *Proc. of the Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, April 1996.
[35] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
[36] A. Meijster, J. Roerdink, and W. Hesselink, "A general algorithm for computing distance transforms in linear time," in *Mathematical Morphology and its Applications to Image and Signal Processing*. Kluwer, 2000, pp. 331–340.
[37] G. Nemhauser and L. Wolsey, Eds., *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
[38] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.
[39] S. van der Zwaan, "Vision based station keeping and docking for floating robots," Master's thesis, Instituto Superior Técnico, Lisbon, Portugal, May 2001. [Online]. Available: vislab.isr.ist.utl.pt
[40] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.
[41] E. Malis and F. Chaumette, "2 1/2 D Visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, June 2000.
[42] J. P. Folcher and M. J. Rendas, "Identification and control of the Phantom 500 body motion," in *Proc. of the Oceans 2001 Conference*, Honolulu, Hawaii, U.S.A., November 2001, pp. 529–535.
[43] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2000.