Proc. of the IEEE Oceans 2002 Conference", TBiloxi, Mississippi, U.S.A., October 2002

Results on Underwater Mosaic–based Navigation

Nuno Gracias, Sjoerd van der Zwaan, Alexandre Bernardino and José Santos-Victor
Instituto Superior Técnico & Instituto de Sistemas e Robótica
Av. Rovisco Pais, 1049–001 Lisboa Codex, Portugal

Abstract—

This paper presents results on mosaic-based visual navigation of an underwater autonomous vehicle, navigating close to the sea floor. A high-quality video-mosaic is automatically built to be used as a representation of the environment. 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 registering the instantaneous views acquired by the vehicle with the mosaic. Extensive testing was conducted at sea, where the vehicle was able to autonomously navigate over the mosaic for extended periods of time. The presented results illustrate the validity and appropriateness of the approach.

I. Introduction

The underwater environment poses a difficult challenge for precise vehicle positioning. One of the contributing reasons is the absence of electromagnetic signal propagation that 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 is the most appropriate.

The methodology described in this paper was devised for mission scenarios where an autonomous platform is required to map an area of interest and to navigate upon it, as illustrated in Fig. 1. First, a high–quality video–mosaic is automatically built and used as a representation of the sea-bottom. Then, a visual servoing strategy is devised to drive the vehicle along a specified trajectory defined over the mosaic map. A possible application scenario for these methods is in underwater archeological site exploration or in marine geological surveys, where an AUV is required to do an initial area mapping followed by periodic inspections.

Email: ngracias@isr.ist.utl.pt. The work described in this paper has been supported by the Portuguese Foundation for the Science and Technology PRAXIS XXI BD/13772/97, and NARVAL Esprit-LTR Proj. 30185.

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: low–rate image-to-mosaic registration and fast motion estimation between consecutive frames of the incoming video stream. Localization information can either be expressed directly in terms of an image frame (e.g. pixel location on the mosaic) or can be 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.

The use of video mosaics as environmental representations for navigation has received considerable attention from both the land and the underwater robotics communities. In the area of land robotics, Kelly[10] addressed the feasibility of using large mosaics for industrial robot guidance. Recent progress has been attained in devising efficient methods to cope with the complexity of mapping large cyclic environments comprising linear segments[14]. In the context of underwater robotics, the use of mosaicing techniques for navigation is a topic of increasing research interest [12], [11], [3]. Xu[16] investigated the use of seafloor mosaics, constructed using temporal image gradients, in the context of concurrent mapping and localization and for real-time applications. Huster[8] described a navigation interface using live-updated mosaics, and illustrated the advantages of using it as a visual representation for human operation. However, as 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 with which this paper more closely relates to is the work of Fleischer[2], in the sense it combines spatially consistent mosaic with underwater ROV navigation. In their approach, the navigation system requires additional sensors to provide heading, pitch and yaw information. This contrasts with our approach, which relies solely upon vision to provide information for all the relevant degrees of freedom during navigation.

The structure of this paper is as follows. Section II.

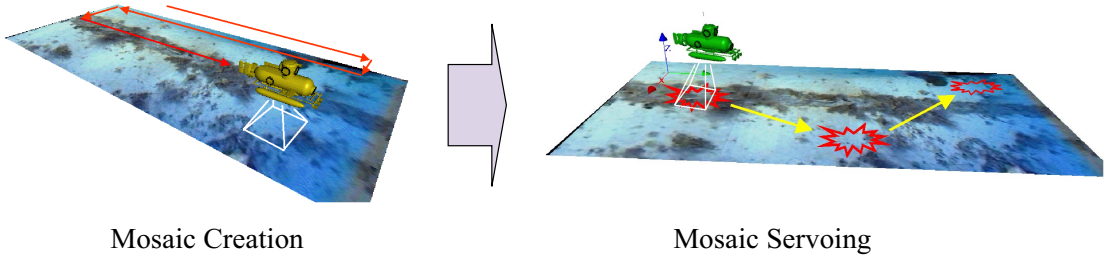

Fig. 1. The operation modes for the implemented mosaic-based navigation system.

gives a brief overview on algorithm used for the creation of mosaic maps. The following Section III. addresses navigation topics, namely the localization and tracking with respect to the mosaic and the trajectory generation. Section IV. presents the implemented visual control scheme. Representative results are presented in Section V., which illustrate the good performance of the overall method. Finally Section VI. presents some discussion and conclusions.

II. Mosaic Map Creation

When mapping large areas, the quality constraints for the mosaicing process are very demanding, given the important role of the mosaic as the only environmental representation. The implemented mosaic creation method, which is summarized in this section, builds upon previous work on underwater video mosaicing[4] and pose estimation[5]. For a detailed description please refer to [6].

The method comprises three major stages. The first stage consists of the sequential estimation of the image motion, using a reduced image motion model. The set of resulting consecutive homographies is cascaded, in order to infer the approximate topology of the camera movement. The topology information is then used to predict the areas where there is image overlap resulting from non-consecutive images.

Secondly, the overall topology is refined by iteratively executing the following two steps. (1) Point correspondences are established between non-adjacent pairs of images that present enough overlap. (2) The topology is updated by searching for the set of homographies that minimizes the overall sum of distances in the point matches.

The final stage of the algorithm consists of estimating the set of homographies and a world plane description that best fit the observation data. As the main concern is attaining high registration accuracy, a general parameterization of the homographies with 6 DOF for the pose is used, which is capable of modelling the effects of wave–induced general rotation and translation.

The overall method is fully automatic and can handle a very general vehicle motion, including loop trajectories, or zig-zag scanning patterns. The method main features are the ability to cope with long image sequences, automatic inference of the path topology and full 3–D recovery of the overall geometry, up to scale.

III. Mosaic Navigation

The mosaic based navigation comprises 3 distinct modules: localization, guidance and control signal generation. An overall block illustration of the main modules is given in Figure 2.

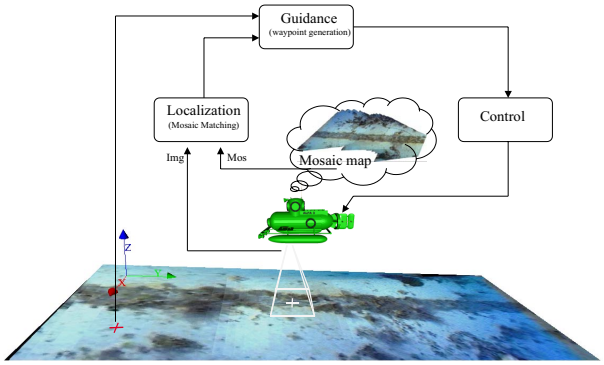

Fig. 2. Overall visual servoing control scheme.

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. In order to do so, a coarse estimate of the vehicle 3D position and orientation, with respect to the mosaic world referential, is required. 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. From this estimate, a corresponding homography is computed and used for searching for point matches.

The online localization comprises two complementary processes which run in parallel, at very distinct rates.

Image–to–Mosaic – The currently acquired image is matched directly over the mosaic, as to have an absolute position estimate. For this, a robust matching procedure is used to filter out correspondence outliers from a set of possible point matches initially found by area correlation. The current position estimate is used to restrict the search area. If the matching is not successful on the first attempt, then a spiral-shaped search pattern is used in the subsequent tries. In each new attempt, the search for matches is car-

ried out using a distance interval from the previous try[1].

Image–to–Image – This process estimates the incremental vehicle motion by matching pairs of images from the incoming video stream. The effectiveness 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 potential time-consuming task because successful mosaic matching might not be achieved at the first attempt. However it 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 on-line tracking is the four-parameter homography, that accounts for 3-D translation and rotation over the vertical axis. 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

To illustrate the usefulness of mosaic as navigation maps, a simple trajectory generation algorithm was implemented. The main purpose of generating trajectories is to make the vehicle avoid the map areas in which 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. For the results in this paper, only the distance to the mosaic borders was considered, but the method can straightforwardly be used to avoid any region definable *a priori* in the map.

[1]In order to find the appropriate intervals, a set of experiments have been conducted using typical underwater images and mosaics. The breakdown of the matching algorithm was evaluated by testing it when coping with increasingly longer intervals.

The first step consists in the creation of a cost map which defines the cost associated with navigating over every elementary region of the map image. In here, the regions to be avoided have higher costs than the rest. This cost map is created by using the distance transform on a reduced–sized binary image of the valid region of the mosaic map. The outcome of this operation is a cost image in which each pixel of the valid mosaic region contains a positive value that decreases with the distance to the border of the valid region. Outside the valid region, the cost is set to a sufficiently high positive number.

Given the current position of the vehicle on the mosaic (*i.e.* the projection of the camera optical axis on the mosaic plane) and the desired end position, we want to find the path that minimizes the accumulated cost over the cost map. This minimization problem can be formulated as a minimal path cost problem, where a path is defined as an ordered set of neighboring locations on the mosaic map. An efficient way to solve it is using Dijkstra's algorithm [13], whose complexity is $O\left(m^2\right)$ where m is the number of pixels in the cost image. An example of the generation of trajectories using this method is presented in Fig. 3.

Fig. 3. Trajectory generation example – Mosaic with superimposed trajectory (left), and corresponding cost image (right).

The computation of the cost image is performed off-line, during the mosaic creation phase. The generation of a new trajectory is required to be performed on-line during the mosaic servoing whenever a new end-point is specified. For the purpose of avoiding the mosaic edges, a relatively small number of trajectory waypoints are 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.

IV. Visual Control

The test–bed ROV used in this paper is equipped with two horizontal back thrusters and a vertical one. The geometric arrangement of these propellers has motivated the design of *decoupled controllers* to operate the robot in the horizontal plane and maintain a constant vertical distance to the floor.

The controllers were designed within the framework of the visual servoing strategies [9]. Therefore, the devised control law makes use of direct image measurements, as opposed to the use of 3–D pose information, which

can be calculated on-line with small additional computational cost. This framework also presents the advantage of easy integration and switching to other visual–based behaviours, such as template–based station keeping [15] which uses the same control approach.

A. Servoing over the Mosaic

Servoing to a goal position over the mosaic is defined as the regulation to zero of an image error function $\mathbf{e} = \mathbf{s} - \mathbf{s_d}$. Here, $\mathbf{s} = (x_c, y_c)$ is the projection of the current image center onto the mosaic, and $\mathbf{s}_d = (x_d, y_d)$ represents the desired docking point, as illustrated in Fig. 4.

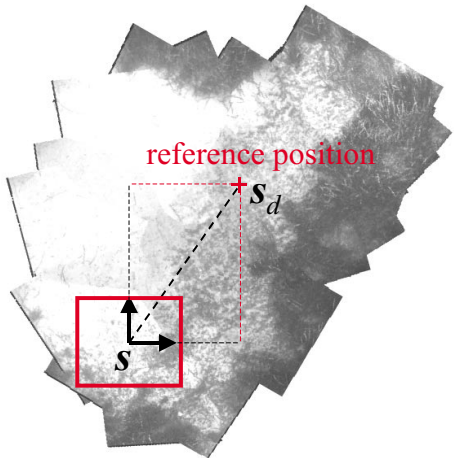

Fig. 4. Definition of error measures on the mosaic. The current image frame is represented by the frame rectangle and the reference is marked by the cross.

Changes in the image features, $\dot{\mathbf{s}}$, can be related to changes in the relative camera pose. This kinematic relationship is often referred to as the *image Jacobian* or the *interaction matrix* [9], [1]:

$$\dot{\mathbf{s}} = \mathrm{L}\mathbf{v_{cam}} \tag{1}$$

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

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

This Jacobian depends both on the image point coordinates and their depth, Z. An exponential decrease of the error function is obtained by imposing $\dot{\mathbf{e}} = -\lambda\mathbf{e}$, with λ some positive constant. Using Eq.(1), we can then solve for the camera motion that guarantees this convergence:

$$\mathbf{v}^*_{\mathbf{cam}} = -\lambda \mathrm{L}(\mathbf{s},\mathbf{Z})^{+}(\mathbf{s} - \mathbf{s_d}) \tag{3}$$

where $\mathbf{v}^*_{\mathbf{cam}}$ 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:

$$\mathbf{v_{cam}} = \mathrm{J_{r2c}}\bar{\mathbf{v}}_{\mathrm{robot}} \tag{4}$$

where $\bar{\mathbf{v}}_{robot}$ contains the controllable velocity components of the vehicle velocity screw and $\mathrm{J_{r2c}}$ is the robot–to–camera Jacobian relationship. This Jacobian is a function of the camera position and orientation in the vehicle reference frame, and can easily be computed from transforming linear and angular velocity components between the frames. With $\mathrm{J_{r2c}}$, it is possible to re-formulate the control objective in terms of desired vehicle velocity components, such that the image center is driven towards the docking point over the mosaic. It also 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 (4) into (1), we obtain an expression that relates the image motion to the vehicle velocity:

$$\dot{\mathbf{s}} = \mathrm{L}\mathrm{J_{r2c}}\bar{\mathbf{v}}_{\mathrm{robot}} \tag{5}$$

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

$$\bar{\mathbf{v}}^*_{robot} = -\lambda\big(\mathrm{L}(\mathbf{s},\mathbf{Z})\mathrm{J_{r2c}}\big)^{+}(\mathrm{s} - \mathrm{s_d}) \tag{6}$$

Figure 5 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.

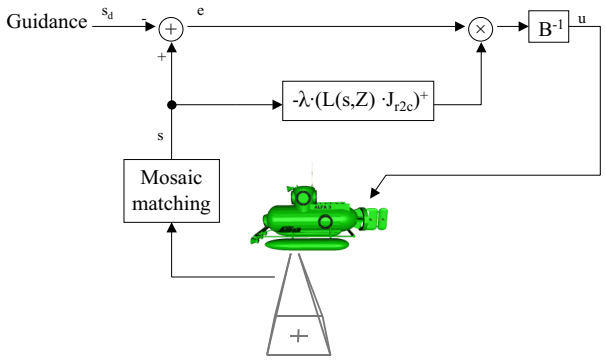

Fig. 5. Control block diagram.

B. Auto–altitude control

The controller for the vertical plane aims at maintaining the camera at constant altitude during navigation. The controller design is such that it maintains the projection of the current image onto the mosaic at the same scale.

To recover the scale, we turn to [7], from which it follows that the image scaling induced by a homography can be

computed from the determinant of its upper left 2×2 block:

$$s = \sqrt{|\mathrm{H}_{2\times 2}|} \tag{7}$$

For the auto-altitude controller, we consider the current image–to–mosaic homography and reconstruct its scaling factor. This is then compared to a reference scaling (typically taken from the initial image–to–mosaic homography) so as to generate the appropriate control error. For auto altitude control, we only consider the camera velocity along its optical axis, whose desired value is then generated proportional to the apparent scale error:

$$\mathbf{u}_{\mathrm{cam}} = -\mathbf{k}(\mathbf{s} - \mathbf{s_d}) \tag{8}$$

Once again, relating Eq. (8) to the vehicle velocity vector through Eq. (4) (where the Jacobian this time is calculated only for the vehicle vertical velocity), results into the desired vertical velocity control law:

$$\mathbf{u}_{\mathrm{robot}} = -\mathbf{k}\mathbf{J}^{+}_{\mathbf{r2c}}(\mathbf{s} - \mathbf{s_d}) \tag{9}$$

where $(s - s_d)$ is the control error J^{+}_{r2c} is a pseudo-inverse of the robot-to-camera Jacobian matrix, taking into consideration the vertical motion of the vehicle.

V. Results

The results reported in this paper were obtained from experiments conducted using a custom modified Phantom 500SP ROV. The ROV is illustrated in Fig. 6 and among other sensors, is equipped with a pan and tilt camera. The controllable degrees of freedom are defined 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 vertically placed thruster. This arrangement creates non-holonomic motion constraints thus requiring the vehicle to undergo complex maneuvers during posture stabilization.

Fig. 6. Computer controlled Phantom ROV with the on–board camera. The camera housing is visible in the lower center, 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. For the servoing tests, the mosaic of Fig. 7 was used. This mosaic is sized 477×544 pixels and was created using 46 images of 192×144 pixels. The images were 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 9.5×10.8 meters.

It should be noted that the mosaic 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.

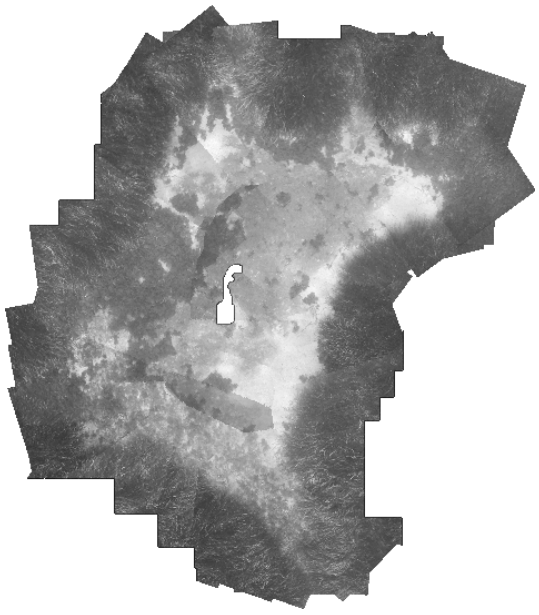

Fig. 7. Fronto–parallel rendering of the mosaic used for the navigation experiments, covering an area of approximately 64 m^2.

An illustrative underwater servoing experiment is presented in Fig. 8 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 a 42 second part of the run is given in Fig. 9.

As stated before, 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

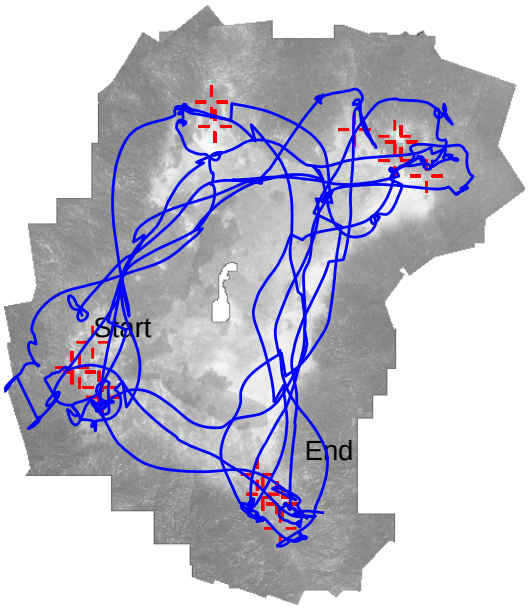

Fig. 8. Underwater mosaic servoing experiment. This plot shows a top–view of the ROV trajectory for the complete run with the reference positions marked with crosses. The ROV trajectory was recovered for the on–line image–to–mosaic matching with updates from the image–to–image tracking, and is marked with the full line.

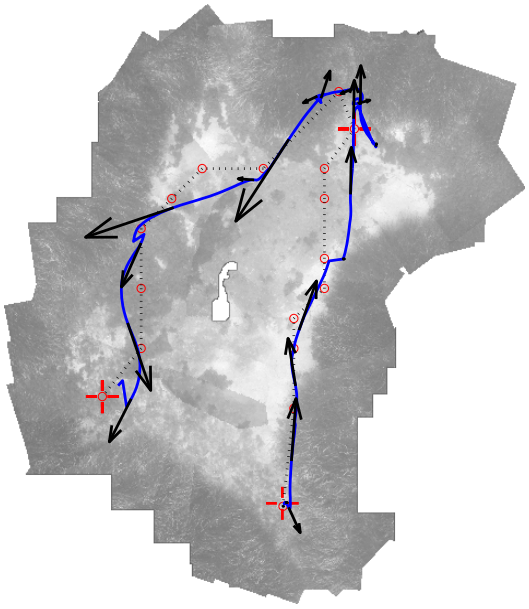

Fig. 9. Trajectory detail comprising two endpoints. The generated path contains a set of waypoints and is marked by the dashed line. In order to allow the sense of speed, a set of arrows is superimposed. The arrows are drawn every 2 seconds and sized proportionally to the platform velocity. The abrupt changes in the trajectory correspond to the position estimate corrections due to the image–to–mosaic matching.

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.

During the 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. It was therefore possible to recreate the trajectory 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 10 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 to the instant when the mosaic matching information becomes available. It can be noticed that 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. 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 taken into account in this set of tests, the magnitude of the accumulated error can be used to adjust the image–to–mosaic matching frequency, thus adapting to cases where the image–to–image tracking performance changes.

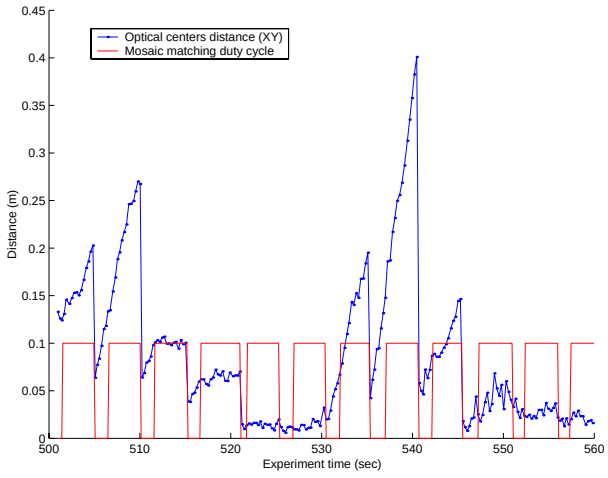

Fig. 10. Diference between the online position estimate, using mosaic matching with inter–image tracking updates, and the offline estimate, obtained by maching all the images over the mosaic. This figure shows the horizontal (XY) distance, where the online mosaic matching instants are marked with small circles.

VI. Conclusions

This paper addressed the problem of underwater autonomous navigation using video mosaics as maps for servoing, where vision is the only sensory modality. The methodology described is devised for mission scenarios where an autonomous platform is required to map an area of interest and navigate upon it afterwards.

Illustrative results from extensive tests at sea were presented, in which a camera-equipped commercial ROV was able to autonomously navigate over automatically created mosaics without getting lost for extended periods of time. The testing conditions illustrated some of the common difficulties for visual based mosaicing and navigation, such as the presence of moving objects and non-planar sea bottom, and served to illustrate the robustness and appropriateness of the approach.

This paper contributes to the field by demonstrating the feasibility of using video mosaics as navigation maps for underwater vehicles moving in 3D, relying on the visual servoing framework. The creation of autonomous vehicles requires the ability to navigate in unstructured and unknown environments, without resorting to external positioning methods. The framework and experimental results of this paper constitute a definite achievement regarding this goal.

References

[1] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[2] Stephen Fleischer. *Bounded–Error Vision–Based Navigation of Autonomous Underwater Vehicles*. PhD thesis, Stanford University, California, USA, May 2000.

[3] 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)*, pages 4010–4015, Washington DC, USA, May 2002.

[4] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79(1):66–91, July 2000.

[5] N. Gracias and J. Santos-Victor. Trajectory reconstruction with uncertainty estimation using mosaic registration. *Robotics and Autonomous Systems*, 35:163–177, July 2001.

[6] N. Gracias and J. Santos-Victor. Underwater mosaicing and trajectory reconstruction using global alignment. In *Proc. of the OCEANS 2001 MTS/IEEE Conference*, Honolulu, Hawaii, U.S.A., November 2001.

[7] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[8] 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 MTS/IEEE Conference*, Nice, France, September 1998.

[9] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[10] A. Kelly. Mobile robot localization from large scale appearance mosaics. *International Journal of Robotics Research (IJRR)*, 19(11), 2000.

[11] S. Negahdaripour and P. Firoozfam. Positioning and image mosaicing of long image sequences; Comparison of selected methods. In *Proc. of the IEEE OCEANS 2001*, Honolulu, Hawai, USA, November 2001.

[12] 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.

[13] G. Nemhauser and L. Wolsey, editors. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[14] R. Unnikrishnan and A. Kelly. A constrained optimization approach to globally consistent mapping. In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 2002.

[15] S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39(3–4):145–155, June 2002.

[16] X. Xu. *Vision–based ROV System*. PhD thesis, University of Miami, Coral Gables, Miami, May 2000.