Using vision for underwater robotics: video mosaics and station keeping

José Santos-Victor, Nuno Gracias, Sjoerd van der Zwaan

Instituto Superior Técnico & Instituto de Sistemas e Robótica
ISR - Torre Norte; Av. Rovisco Pais, 1
1049-001 Lisboa; PORTUGAL
{jasv,ngracias,sjoerd}@isr.ist.utl.pt

Abstract— ***In this paper we discuss the use of vision for underwater vehicles. The work described here has been developed in the context of the European Research Project NARVAL - Navigation of Autonomous vehicles via Active Environmental Perception (ESPRIT-LTR project 30185), aiming at designing and implementing reliable navigation systems of limited cost for mobile robots in unstructured environments, without the use of global positioning methods. In this paper, we will focus on two main applications of computer vision in the context of underwater robotics: i) building and using video mosaics of the sea bottom and ii) vision-based control for station keeping and docking. We describe our approach to these tasks and present results obtained during sea experiments.***

I. INTRODUCTION

The use of visual information for underwater vehicles has attracted considerable attention in the past few years. The reasons for this interest are multi-fold. On one hand, vision is a high-resolution sensing modality that provides information about the surrounding environment at high bandwidth. On the other hand, the understanding of the multi-view geometry, availability of robust algorithms and the necessary computational power have made a number of real-time applications possible.

Visual control loops and representations can be introduced in order to increase the flexibility and the accuracy of underwater vehicles. In this paper, we described two applications of vision in the context of underwater vehicles, illustrating both visual representations and vision-based control approaches. The experiments were obtained with a Phantom ROV, with an on-board pan-tilt camera and an off-board personal computer where all signal processing is done.

A first application consists in building high quality Video Mosaics of the sea bed from long image sequences, and estimating the camera trajectory. The camera/vehicle motion can be very general, including loop trajectories, or zig-zag scanning patterns. The method comprises three major stages. Firstly, the image motion is computed in a sequential manner, with simple motion models, to create a set of consecutive image transformations (usually called homographies), which are cascaded in order to infer the approximate topology of the camera movement. Secondly, a motion refinement is performed, by iteratively executing the following two main steps. (1) Point correspondences are established between non-adjacent pairs of images that present enough overlap. This is a time consuming operation, alleviated by the use of prior information about the location of the image correspondences, computed in the first stage. (2) The topology is refined, by searching for the set of homographies that minimizes the overall sum of distances in the point matches. Finally, a global minimization is carried out, using the most general 6-degree of freedom motion model and a cost function based on the errors of the point matches between all the images. The minimization process searches for the best set of pose parameters (describing the 3D pose of the camera) and for the best fitting description of the world plane

The second aspect described in the paper is that of automatic Vision-based Sstation Keeping, relative to some visual landmark. The vision based station keeping task is defined locally in the neighborhood of some visual landmark and consists of stabilizing the vehicle relative to this landmark, rejecting external disturbances. For underwater robots, staying fixed at some given position is not inherent since it is susceptible to significant drift. Station keeping is therefore an important behavior for tasks such as underwater inspection and manipulation.

A selected image region is used as a visual landmark, whose temporal changes, induced by the vehicle's motion, are tracked. Our tracking system is both fast and accurate and thus adequate for real-time implementation. It determines camera motion from the registration between the current live image and an initial reference image. For a camera moving in 3D, the exact image motion model is a planar projective transformation, which can be parameterized by 8 parameters and capture all possible deformations in the image plane. We avoid an exhaustive search on the parameter space by using a set of motion models that sample the search space for expected image deformations. To enhance robustness, the history of past detected motions is iteratively substituted in the set of motion models, thus predicting future deformations. This also provides a means to monitor and identify the principal displacements of the underwater robot moving in 3D space. In addition, we use optic flow information to provide the tracker with an initial estimate of the current transformation parameters. Finally, the problem of automatic landmark selection is addressed.

The tracking information is then used to synthesize the station keeping controller. The control objective is to drive the ROV back to the desired view under external disturbances. The main difficulties are related to the vehicle's motion constraints, having a limited number of controllable degrees of freedom. To add robustness, an image stabilization technique is applied that automatically controls the camera's pan and tilt degrees of freedom so as to keep the visual landmark constantly in view during maneuvers.

II. Background

For both the video mosaicking application and the visual station keeping task, we assume that the sea bottom can be locally approximated by a planar surface. With such an approximation we can parameterize the image motion and design robust estimation methods that would otherwise be unfeasible. In this section we will assume the reader to be familiar with the basic concepts and properties of projective geometry [17].

A. *Camera model*

The camera model used in this paper is the standard pin-hole model, which performs a linear projective mapping of the 3D world into the image frame. We also assume that the camera calibration has been performed beforehand, and that the 3×3 matrix K containing the intrinsic parameters has been estimated [32], [25]. With the pin-hole model, planar image motions cannot be adequately modeled by simple transformations, like affine or translational. A projective planar transformation is the exact motion model when a camera rotates about its eyepoint or if the imaged surface is planar.

B. *Planar projective transformations*

The 2D projective transformation is represented by a 3×3 homography, H. This transformation maps image points, such that $\mathbf{x}' = H\mathbf{x}$, where $\mathbf{x}'$ and $\mathbf{x}$ are the homogeneous coordinates of the image points (x', y') and (x, y), respectively. This transformation is defined up to a scale factor and therefore has eight degrees of freedom, given by the entries of H. Usually, these transformations, are parameterized as a function of a vector $\mathbf{q}$. It follows [14] that this homography can be decomposed into a hierarchical chain of transformations in the image plane:

$$H = H_s H_a H_p = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{v}^T & \lambda \end{bmatrix} \qquad (1)$$

where H_s is a scaled Euclidean transformation, having 4 d.o.f that account for translation, rotation and scaling in the image plane, H_a affects affine properties with K as a 2 d.o.f. upper-triangular matrix normalized as detK = 1, containing the shear and aspect ratio parameters, H_p is a 2 d.o.f. transformation that accounts for projective distortion, as specified in the parameter vector $\mathbf{v}^T$ and λ is a positive scale factor . These degrees of freedom are illustrated in Fig. 1 and define a more intuitive parameterization for the transformation rather then the entries of H. This parameterization is such that a zero valued parameter vector specifies the identity transform. The computation of a planar transformation requires at least four pairs of corresponding points. In the case of more than four correspondences, a straight-forward least-squares linear estimation can be accomplished[21].

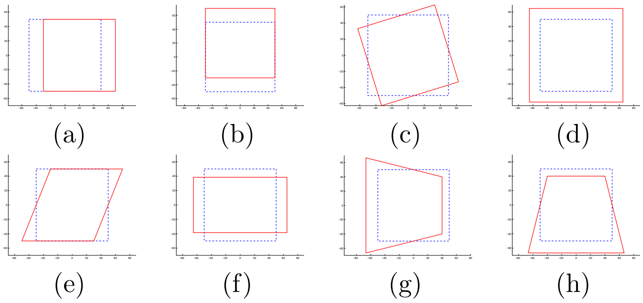


Fig. 1. Degrees of freedom of the planar projective transformation on images: (a) translation along the horizontal image axis, (b) translation along the vertical image axis, (c) rotation, (d) scaling, (e) shear, (f) aspect ratio, (g) projective distortion along the horizontal image axis, (h) projective distortion along the vertical image axis

C. *Image registration*

Given a reference image or *template* T and a target image I, the image registration problem is defined as computing a transformation that relates points (x', y') in the template image to points (x, y) in the current target image. Usually, these transformations, are parameterized as a function of a vector $\mathbf{q}$, such that $(x', y') = \mathcal{H}_{\mathbf{q}}(x, y)$. This transformation is on image coordinates and therefore defines an image warping that maps pixel intensity values from the template image T to the current target image I:

$$\mathcal{W}(\mathbf{q}, T) \mapsto I$$

Here, $\mathcal{W}(\mathbf{q}, T)$ specifies the image warping according to the transformation parameters $\mathbf{q}$.

To register the current image with the template, the best possible match can be obtained through the minimization of an error function, using an appropriate norm, such as the sum-of-squared-differences ($L2$-error criterion). Writing images as column vectors, the estimate of the current transformation parameters at each time step is then found as:

$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \big(\frac{1}{2} \parallel I - \mathcal{W}(\mathbf{q}, T) \parallel^2 \big) \qquad (2)$$

When iteratively tracking an image region through a video sequence, at each time instant, an initial guess of the current transformation parameters is given by the parameters of the previous step. This provides a first step towards the solution so that only small adjustments remain to be made. In such a scheme, an approximate error criterion is given by:

$$\Delta\hat{\mathbf{q}} = \arg\min_{\Delta\mathbf{q}} \big(\frac{1}{2} \parallel \mathcal{W}^{-1}(\mathbf{q_0}, I) - \mathcal{W}(\Delta\mathbf{q}, T) \parallel^2 \big) \qquad (3)$$

where $\mathcal{W}^{-1}(\mathbf{q_0}, I)$ is the image obtained from the inverse warp that maps the current image I approximately onto the template T, according to the initial guess $\mathbf{q_0}$. Upon minimizing this criterion, we look for the best residual warp, $\mathcal{W}(\Delta\mathbf{q}, T)$ that accounts for the observed difference between the image $\mathcal{W}^{-1}(\mathbf{q_0}, I)$ and the template T. The current transformation parameters are then updated according to:

$$\hat{\mathbf{q}} = \Delta\hat{\mathbf{q}} \otimes \mathbf{q_0} \qquad (4)$$

where $\otimes$ stands for the update operator, which in the case of planar projective transformations corresponds to matrix multiplications of the corresponding homographies.

D. Scaled Euclidean reconstruction

Given an inter-image homography, it is possible to reconstruct the relative displacement of the camera in 3D space, up to a scale factor. This is also known as *scaled Euclidean reconstruction* and allows to reconstruct the relative camera trajectory from image registering through a monocular video sequence. This decomposition is described in [18], relating the homography matrix H with the camera rotation, translation and the world plane which induces the homography. The decomposition is the following

$$H_{21} = K \left(R_{21} + n_1 \frac{t^T}{d_1} \right) K^{-1} \qquad (5)$$

where R_{21} and t are, respectively, the 3×3 rotation matrix and the 3×1 translation vector relating the two 3-D camera frames. The world plane is accounted for through 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, measured along the optical axis.

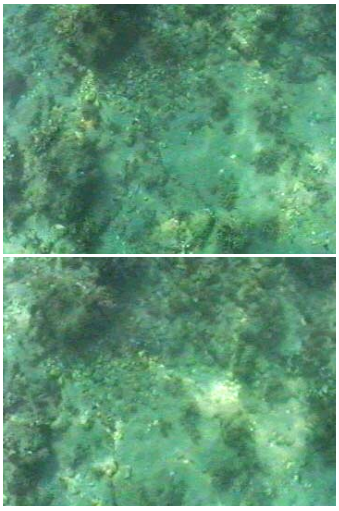


Fig. 2. Two sequential frames, illustrating the difficulty of the matching process for images of very shallow waters, where the lighting conditions change rapidly.

The problem of recovering the motion parameters from an homography for an intrinsically calibrated camera is discussed in-depth by Faugeras[18]. In the most general case there are eight different sets of solutions. However, only two are feasible if one considers the world plane to be non-transparent. These two solutions can conveniently found by means of the SVD decomposition of $M_{21} = K^{-1} H_{21} K$, as presented by Triggs[31].

III. Video Mosaics of the Sea Floor

The basic assumptions for mosaic creation are that the sea bottom is essentially flat, static and without strong illumination changes. This is seldom the case in underwater mapping applications, especially in shallow waters. However, the use of robust estimation over point feature matching greatly alleviates these assumptions and allows for the correct recovery of image motion. As an illustrative example, Figure 2 contains two consecutive frames of an image sequence used in this work which were successfully matched.

Our method comprises three major stages. Firstly, the image motion is computed in a sequential manner, using a simple image motion model, in order to create a set of consecutive homographies. Secondly, a motion refinement is performed. Finally, a global minimization is carried out, using the most general 6-degree of freedom motion model and a cost function based on the errors of the point matches between all the images.

A. Initial Motion Estimation

The first part of the algorithm consists on the sequential estimation of inter-frame homographies.

For each image I_k, a set of point features, corresponding to textured areas, is extracted using the Harris corner detector[24]. The features are then matched directly on the following image I_{k+1}, using correlation-based procedure, from which two lists of corresponding points are obtained.

Due to the error prone nature of the matching process, it is likely that a number of points will be mismatched. Therefore, a robust estimation technique is required 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}. In this paper, a variant of LMedS with random sampling[21] was used for minimizing the median of sum of the square distances,

$$\arg\min_i \Big(\mathrm{med}\big\{ d^2\big({}^{(k)}x_i, T_{k,k+1} \cdot^{(k+1)} x_i\big) \\ + d^2\big({}^{(k+1)}x_i, T^{-1}_{k,k+1} \cdot\ {}^{(k)}x_i\big) \ \big\} \Big) \tag{6}$$

where $d\left(\cdot,\cdot\right)$ stands for the point-to-point Euclidean distance, and ${}^{(k)}x_i$ is the location of the i^{th} feature extracted from image I_k and matched with ${}^{(k+1)}x_i$ on I_{k+1}. The minimizing algorithm works by randomly sampling sets of points with the minimum number of matches required for the linear computation of H. The set that minimizes the cost function is selected and the homography is re-estimated using simple least-squares with all matches whose distance are below a specified error limit. The image matching is considered successful if the number of matches used in the final least-squares estimation is sufficiently high.

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 applications it is very common for the image acquisition rate to be high when compared to the camera motion. This results in very high overlapping between consecutive frames that convey redundant information. In the work presented, a selection criteria was used to selected a subset of frames, thus reducing the memory and processing requirements for the next stages. The frames are selected such that their superposition is the smallest above a given minimum acceptable overlap percentage. This threshold insures the ability of the selected images to be correctly matched, and is chosen based on the results of preliminary matching trials.

B. Iterative Motion Refinement

After the initial motion estimation step, every image in the reduced sequence can be spatially related with any other image, by appropriately cascading the homographies. Possible overlap between non-consecutive images can be predicted, and used for searching new image matches.

In this stage, the topology is refined 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 per image (2D translation and rotation), that implicitly assumes the camera is facing the ground and keeps a constant distance. The reason behind the choice of a simpler motion model for the first two stages of the algorithm, has to do with the effectiveness of the topology inference. Alternatively, one could have resorted to the use of the most general 8-parameter homographies, as this is the only model that can cope with general perspective distortion *and* allow for fast linear estimation. However, it has more degrees of freedom than 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.*,

$$F(X,\Theta) = \sum_{i,j} \sum_{n=1}^{N_{i,j}} [d^2\left(x_n^i, H(\Theta_i,\Theta_j)\cdot x_n^j\right) \\ + d^2\left(x_n^j, H^{-1}(\Theta_i,\Theta_j)\cdot x_n^i\right)]$$

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. The minimization is carried out using a non-linear least squares algorithm[29]. The cycle of matching and topology refinement is executed until no new image pairs can be matched.

In order to speed-up the optimization procedure (and, thus, the motion refinement cycle time), a sub-mosaic aggregation scheme was implemented and tested. 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 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. Trajectory Estimation

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. Bearing this in mind, a parameterization was chosen in which all the camera pose 6 degrees of freedom are explicitly taken into account. This has also the additional advantage of allowing the camera path to be recovered during the process.

Furthermore, the estimation of the homographies for this model does not impose, *per se*, the condition of a single world plane from which the homographies are induced. This condition can be imposed by augmenting the overall estimation problem with additional parameters that describe the position and orientation of the world plane. The world plane description must then be included on the parameterization of the homographies.

The adopted parameter scheme is the following. One of the camera frames is chosen (usually the first) 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[1], measured along the Z-axis. The homography relating frames i and j is

$$H(\Theta_p, \Theta_i, \Theta_j) = K \cdot \left(R\left(\Theta_i\right) + n\left(\Theta_p\right) \cdot t^T\left(\Theta_i\right)\right) \cdot$$

$$\cdot \left(R\left(\Theta_j\right) + n\left(\Theta_p\right) \cdot t^T\left(\Theta_j\right)\right)^{-1} \cdot K^{-1}$$

where Θ_i and Θ_j are pose vectors containing 3 rotation angles and 3 translation values with respect to the reference frame, $R\left(\Theta_i\right)$ 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, and $n\left(\Theta_p\right)$ is the 3-vector with the outward plane normal. The pose vector for the reference camera is the null 6-vector.

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.*,

$$F(X, \Theta) = \sum_{i,j} \sum_{n=1}^{N_{i,j}} [d^2\left(x_n^i, H(\Theta_p, \Theta_i, \Theta_j) \cdot x_n^j\right)$$

$$+ d^2\left(x_n^j, H^{-1}(\Theta_p, \Theta_i, \Theta_j) \cdot x_n^i\right)]$$

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 Equation (5). 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. *Mosaic results*

Extensive testing was conducted in order to evaluate the performance of the algorithms. The image sequences for the results shown in this paper were acquired by a Phantom ROV during a NARVAL Project sea trial, in Villefranche-sur-mer in France. The ROV is equipped with a Sony pan-and-tilt camera, facing the sea floor. It is mounted in the center of a spherical glass housing which induces very little image distortion.

The camera calibration was performed under water using a standard calibration grid and the method described by Heikkilä in [25].

The first sequence refers to a flat sandy area, fully surrounded by algae. During the acquisition, the vehicle was manually driven to follow a zig-zag trajectory that covered most of the area. The sequence comprises 1000 images, corresponding to 400 seconds of video. After the initial matching, a set of 129 images was selected using the criterion of minimal overlap above 50%, which resulted in an average overlap of 54.4%. The mosaic obtained from the last stage of the algorithm, in shown in Figure 3. It was created by choosing the contributing points which were located the closest to the center of their frames. This rendering method is useful when

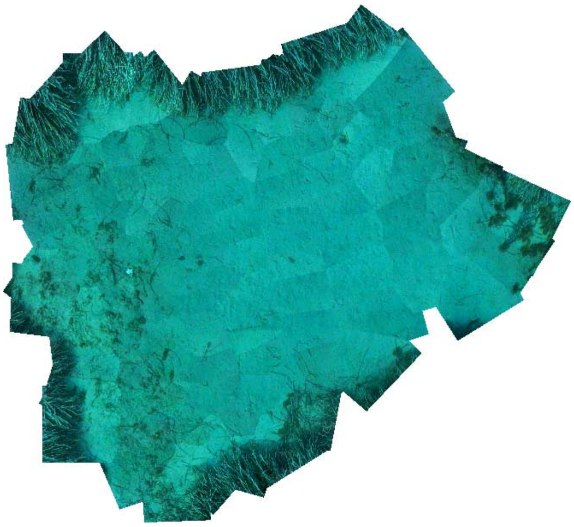


Fig. 3. Final mosaic for the first image sequence. It was created using 129 images selected from the original set of 1000 and rendered with the *closest* operator. The seafloor area covered is approximately 42 m^2.

[1] 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.

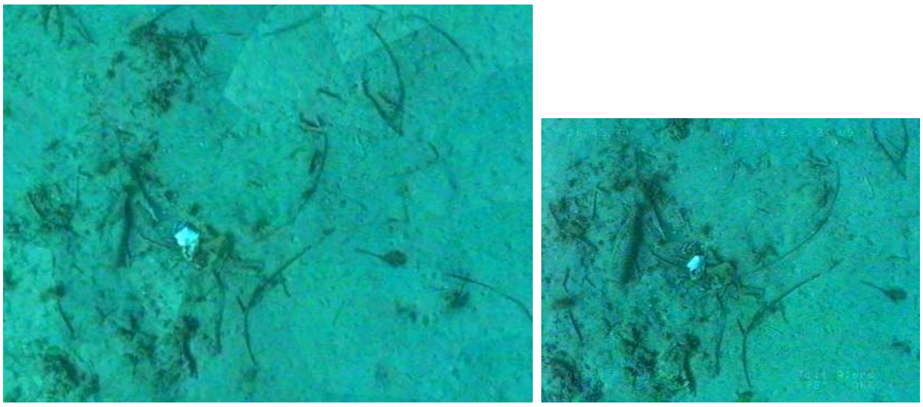


Fig. 4. Area detail of the mosaic for the first sequence (left), and one of the original images (right).

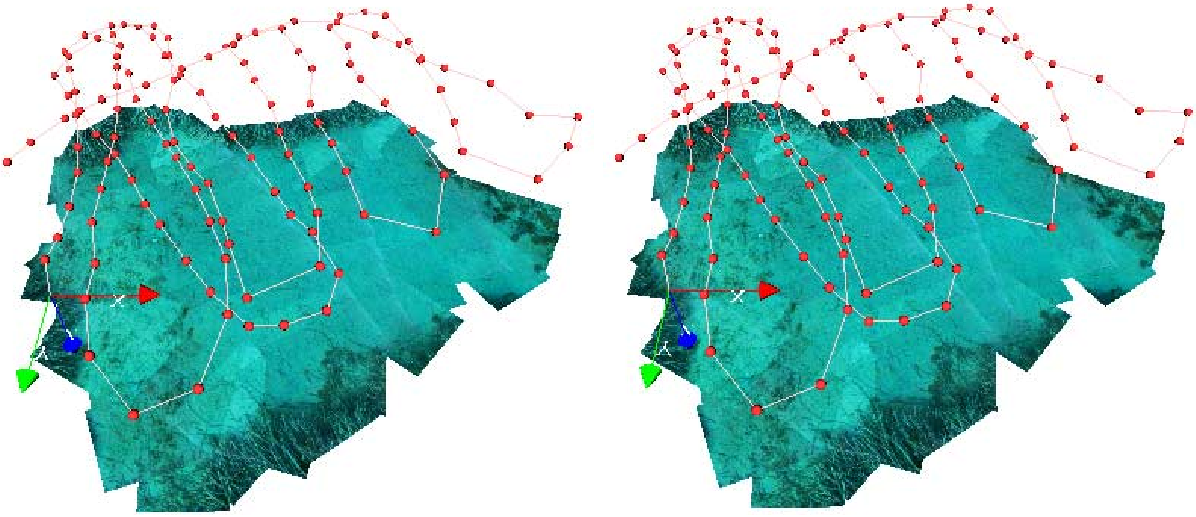


Fig. 5. VRML rendition of the camera path and mosaic for the first image sequence. The world referential is illustrated by the system of axis, which is coincident with the first camera frame. The two views are arranged for crossed eye fusion.

creating the mosaics for navigation and mosaic-based localization. For the cases where the illumination changes are *not* strong, it compares favorably with other commonly used rendering methods, such as the average or the median. This is due to the fact that it better preserves the textures and minimizes the effects of barrel distortion, which tends to be larger near the image borders. A small section of the rendered mosaic is displayed in Figure 4 along with one of the original frames for the same area. The quality of the registration can be assessed from the fact that the visual features (such as small algae leaves) are not disrupted along the visible boundaries of the contributing images. [hbtp]

The second sequence was acquired in very shallow waters, of less than 2 meters in depth, where the effect of sunshine refracted from the surface is clearly noticeable. The vehicle followed a circular trajectory of several turns around a square shaped rock. The original sequence contains 895 images from which 85 where selected using a 60% minimum overlap. The resulting mosaic is presented in Figure 7.

IV. VISION BASED STATION KEEPING

For vision based station keeping, we start by addressing the tracking problem before discussing the use of this information for visual control purposes.

A. *Tracking of image regions*

The tracking system for the station keeping controller aims at tracking a naturally textured landmark in the image plane, whose positional information is then used to tune the station keeping controller.

For tracking, we include optic flow information in a prediction phase by adjusting an affine model to the observed image motion. The affine motion estimate is computed from the temporal and spatial derivatives in the current and previous live images [11], [12]. The advantages are two-fold: (i) by adding information to the

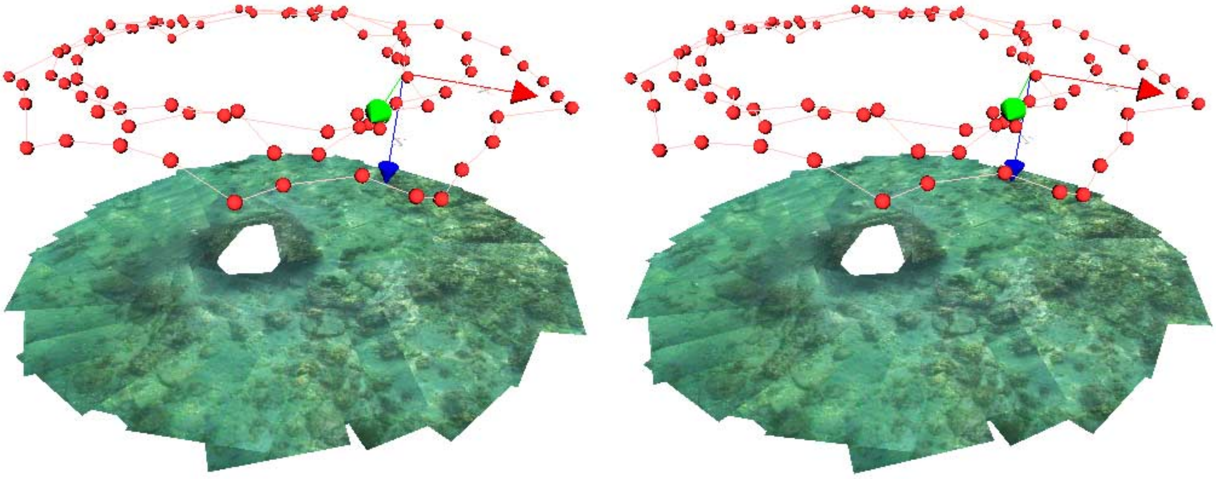


Fig. 6. VRML rendition of the camera path and mosaic. The world referential is illustrated by the system of axis, which is coincident with the first camera frame.

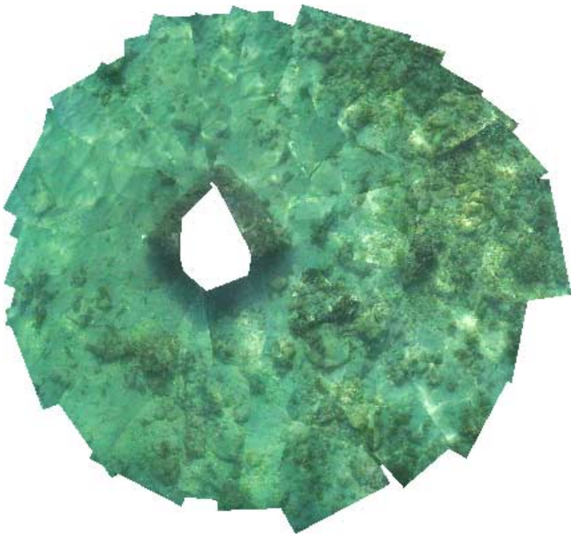


Fig. 7. Final mosaic for the second sequence.

initial guess, the residual transformation parameters are kept small (ii) optic flow provides a means to keep track of the transformation parameters when the visual landmark gets out of the image. Furthermore, it provides a way of monitoring the residual matching procedure, since this solution should be in the small neighborhood of the affine flow prediction.

To find the best residual warp at each time step, we minimize the error function in (3), using a set of m motion vectors $\{\Delta\mathbf{q}_i : i \in (1 \dots m)\}$ that sample the parameter space for expected image deformations. Each motion model, $\Delta\mathbf{q_i}$, transforms the template image T into an image $\mathcal{W}(\Delta\mathbf{q_i}, T)$ that contains image deformations expected to be observed over time. In our implementation, the algorithm samples into the directions of the individual parameters of the transform parameterization, over varying ranges.

The residual transformation parameters that are looked for, $\Delta\mathbf{q}$, can be expressed as a linear combination of the various motion models, $\Delta\mathbf{q}_i$:

$$\Delta\mathbf{q} = \sum_{i=1}^{m} k_i \Delta\mathbf{q}_i \tag{7}$$

The image warping operator can now be considered to be specified by the parameter vector $\mathbf{k} = [k_1 \dots k_m]^T$. The new parameterization is given by:

$$\mathcal{W}(\mathbf{k}, T) = \mathcal{W}(\sum_{i=1}^{m} k_i \Delta\mathbf{q}_i, T) \tag{8}$$

where $\mathcal{W}(\mathbf{k}, T)$ is the image obtained from warping the template T according to the linear combination of motion vectors $\Delta\mathbf{q}_i$. Substituting (8) into the error function (3), the matching problem can be formulated as finding the linear combination of motion vectors that best accounts for the observed difference between the approximately registered current image and the template:

$$\mathbf{k} = \arg\min_{\mathbf{k}} \left(\frac{1}{2} \parallel \mathcal{W}^{-1}(\mathbf{q_0}, I) - \mathcal{W}(\mathbf{k}, T) \parallel^2\right) \tag{9}$$

The image $\mathcal{W}(\mathbf{k}, T)$ is in general a complex and highly non-linear function of the transformation parameters and the texture map defined in the template image. For small deviation around $\mathbf{k} = 0$, it and can be approximated by:

$$\mathcal{W}(\mathbf{k}, T)\Big|_{\mathbf{k}=0} \approx T + B\mathbf{k}$$

Substituting this approximation into the error function in (9), a least square solution can be computed for $\mathbf{k}$:

$$\mathbf{k}_{LS} = (B^T B)^{-1} B^T D \tag{10}$$

where we have introduced $D = \big(\mathcal{W}^{-1}(\mathbf{q_0}, I) - T\big)$ as the observed difference between the approximately registered current image and the template image. After determining $\mathbf{k}$, the solution for $\mathbf{\Delta q}$ can be calculated from equation (7).

Most computational requirements go out with the computation of the pseudo-inverse, $(B^TB)^{-1}B^T$, which can be calculated off-line since it is constructed from the set of motion models and the template image. The only on-line computation is the calculation of the difference image, D, implying an image warp $\mathcal{W}^{-1}(\mathbf{q_0}, I)$. This makes the method very well-suited for real time tracking applications.

With this algorithm, we were able to successfully track a visual landmark undergoing planar projective transformations. A 15 Hz tracking frequency is reached for images with a 128 × 192 pixel size, using an off-the-shelf 450Mhz processor. Fig. (8) shows results of tracking an image region in submarine images. The initially selected image region is used as a template, whose temporal deformations are tracked over time.

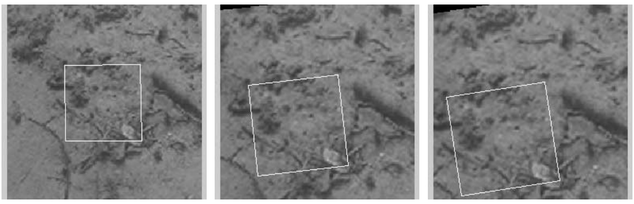


Fig. 8. Tracking an image region in a submarine video sequence.

Another advantage of the difference template method is the ability to customize the set of motion models according to the kind and range of expected image deformations. The choice of the motion models greatly determines the performance of the algorithm. Ideally, this choice should be adapted to the camera motion. This idea has been explored in our implementation of the tracker system, where we include new motion models according to the history of past detected, incremental updates of the transform parameters. In the image plane, these updates point out into the direction and range of expected incremental deformations in near future. An additional small subset is added to the already existing set of motion models and is iteratively adapted to the camera motion. Maintaining the original set intact prevents the algorithm from loosing its ability to sample for deformations in all directions.

When iteratively substituting motion models, new difference templates need to be included into the partial derivatives matrix, B, implying on-line calculation of its pseudo-inverse $(B^TB)^{-1}B^T$. To avoid this, we take advantage of the information already stored in the pre-calculated pseudo-inverse and update it according to the substituted difference image.

In order to characterize the maximum range over which the algorithm is able to accurately estimate inter-image transformations, image motion is simulated from warping an image according to a pre-defined trajectory of the transformation parameters, so that ground-truth information is available. The plot in Fig. (9) shows the results of tracking a reference point on the landmark, in the presence of increasing incremental motion in the image plane, according to a smooth trajectory. It follows that upon iteratively substituting motion models, the algorithm is able to track over a much wider range.

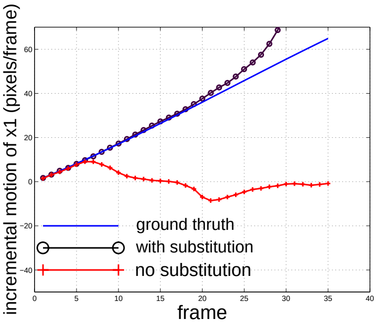


Fig. 9. Tracking the position of a corner coordinate of a selected image patch, in the presence of increasing inter-image motion.

B. *Optimal landmark selection*

When selecting an image region as a template for tracking, its texture map should contain sufficient information so that expected image deformations over time can be observed from it. To automatically select a template from an image, some optimality criterion needs to be evaluated, that takes the observability with respect to the motion models into account.

To do so, we follow the approach in [2], and model the observed difference, $D = \mathcal{W}^{-1}(\mathbf{q_0}, I) - T$, as a linear combination of the pre-calculated difference images, in the presence of additive noise:

$$D = B\mathbf{k} + u \tag{11}$$

where u is additive noise, $\mathbf{k}$ represents the real transformation parameters that is looked for and B is the partial derivatives matrix containing all difference images. The least-square estimate for $\mathbf{k}$ is given in (10) and can be rewritten using (11) as:

$$\mathbf{k}_{LS} = \mathbf{k} + \big((B^TB)^{-1}B^T\big)u \tag{12}$$

In order to have $\mathbf{k}_{LS}$ as a reliable estimate of $\mathbf{k}$, we would like to choose a B, such that the uncertainty introduced by $\big((B^TB)^{-1}B^T\big)u$ is minimized. The partial derivative matrix B is a function of the selected template texture and the set of motion models. For the same set of motion models, different templates result in different values of uncertainty.

To measure this uncertainty, we take the $L2$-norm on the error in the reconstructed signal:

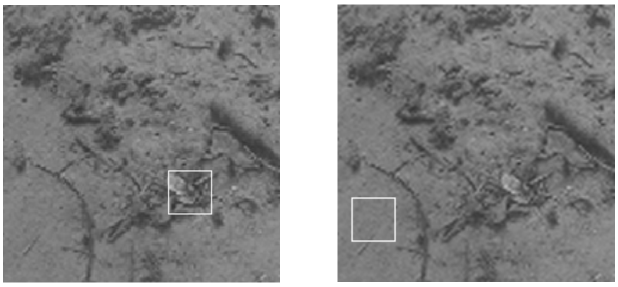
$$\|\mathbf{k}-\mathbf{k_{LS}}\|^2 = \|\big((B^TB)^{-1}B^T\big)u\|^2 \tag{13}$$

Assuming zero-mean, unit variance white noise for u, we can take the expected value of (13), which can be computed as:

$$E\{\|\big((B^TB)^{-1}B^T\big)u\|^2\} = trace\big((B^TB)^{-1}B^T\big) \tag{14}$$

The optimal template is then found by minimizing the expected value of (14), given the set of motion models.

Fig. (10) shows the most and less informative template in an underwater image, for a fixed size landmark. These were found by performing an exhaustive search over the image space.



(a) (b)

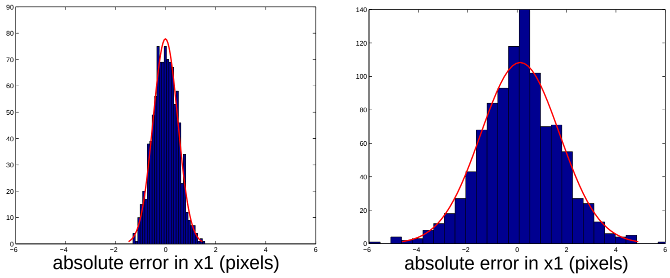
Fig. 10. Automatic landmark selection: (a) most informative image region , (b) less informative image region.

Apart from selecting the most informative window in an underwater image, the minimum value of the expected uncertainty can also be used to set an absolute threshold on images, which can be evaluated to verify whether or not the image contains sufficient information for tracking.

The selection of informative landmarks has a noticeable impact on the tracking accuracy. Some test were performed that evaluated the tracking error on images that contain randomly applied deformations with superimposed image noise. The error is defined as the difference between the real and estimated position of image points and is evaluated for a 1000 trials for both the most informative and the less informative template, as given in Fig. (10). The results are plotted in Fig. (11) and show that sub-pixel accuracy is obtained when tracking informative image regions.

C. Visual station keeping controller

For station keeping, we assume that the ROV is hovering parallel to the ocean floor, having the camera looking approximately perpendicular to a planar region. A decoupled control design is adopted, which station keeps the ROV in the horizontal plane w.r.t the landmark, while maintained at a fixed depth in the vertical plane. Both controllers are formulated in an image based visual servoing framework [10], [9], so that error signals are defined directly in terms of image features. Although defined in the image plane, the task is represented by a particular alignment in 3D-space between the camera/vehicle and the planar landmark.



(a) (b)

Fig. 11. Tracking error for the x-coordinates of the upper-left landmark corner under randomly generated image deformations. Maximum inter-image deformations are in the range of 5 pixels and images where corrupted with zero mean Gaussian noise with 10% standard deviation. The error is evaluated over a 1000 trials for both the most informative template (a) and the less informative template (b).

The image based station keeping task is defined as the regulation to zero of an image error function $\mathbf{e}(\mathbf{s}) = \mathbf{s} - \mathbf{s_d}$, where $\mathbf{s}$ is the image feature parameter vector and $\mathbf{s_d}$ the desired value. The centroid of a tracked image region is used as a feature, whose desired position is at the image center. The image error function is then given by $\mathbf{e} = [x_c, y_c]^T - [x_d, y_d]^T$ and the controller aims at driving the centroid towards the image center under external disturbances like currents.

Changes in the image features 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* [10], [9]:

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{v}_{cam} \tag{15}$$

Where L is the image Jacobian and $\mathbf{v}_{cam}$ is the 6×1 camera velocity screw. The image Jacobian for the centroid is given by the motion field:

$$\begin{bmatrix}\dot{x_c}\\ \dot{y_c}\end{bmatrix} = \mathbf{L}\mathbf{v}_{cam}, \tag{16}$$

where

$$\mathrm{L} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x_c}{Z} & x_c y_c & -(1+x_c^2) & y_c \\ 0 & -\frac{1}{Z} & \frac{y_c}{Z} & (1+y_c^2) & -x_c y_c & -x_c \end{bmatrix} \tag{17}$$

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 (16), we can then solve for the camera motion that guarantees this convergence:

$$\mathbf{v}_{cam}^{*} = -\lambda \mathrm{L}(\mathbf{s}, Z)^{+}(s - s_d) \tag{18}$$

Where $\mathbf{v}_{cam}^{*}$ is the resolved camera velocity that drives the centroid to the image center and L^{+} is the pseudo-inverse of the image Jacobian.

The ROV control inputs are in general defined in the vehicle reference frame, commanding components of the vehicle velocity vector. It is therefore useful to relate the controllable components of the vehicle velocities to camera velocities. This relationship is given by the control input Jacobian:

$$\mathbf{v}_{cam} = \mathrm{J}_{rov}\bar{\mathbf{v}}_{rov} \tag{19}$$

Where $\bar{\mathbf{v}}_{rov}$ contains the controllable velocity components of the vehicle velocity screw and J_{rov} is the control input jacobian. This jacobian is a function of the camera position and orientation in the vehicle reference frame, $\mathrm{J}_{rov} = f\big({}^{rov}R_{cam}, P_{cam}\big)$ and can be easily computed from transforming linear and angular velocity components between the frames. For station keeping, we consider the linear and angular velocity of the vehicle in the horizontal plane, $\bar{\mathbf{v}}_{rov} = [v, \omega]^T$, which are both controllable from the two back thrusters. Substituting (19) into (15), an expression is obtained that relates image point velocities to the vehicle velocity:

$$\dot{\mathbf{s}} = \mathrm{L}\mathrm{J}_{rov}\bar{\mathbf{v}}_{rov} \tag{20}$$

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

$$\bar{\mathbf{v}}_{rov}^{*} = -\lambda\big(\mathrm{L}(\mathbf{s}, Z)\mathrm{J}_{rov}\big)^{+}(K_p\mathbf{e} + K_d\dot{\mathbf{e}} + K_i\int \mathbf{e}dt) \tag{21}$$

Here we have included a PID control action on the image error for dynamic compensation. This expression takes the vehicle motion constraints into account, resulting into trajectories that are physically executable.

D. Visual auto-depth controller

The controller for the vertical plane aims at maintaining the ROV at a fixed depth during station keeping maneuvers. The controller design is such that it maintains the appearance of the landmark in the image plane at the same scale. Having the ROV hovering parallel to a planar region, the scale in the image plane of some selected landmark has a direct physical interpretation in terms of relative depth.

To recover the scale in the image plane, we turn to (1) and rewrite it as:

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & \lambda \end{bmatrix} \tag{22}$$

Where A is a non-singular matrix given by $A = sRK + \mathbf{t}\mathbf{v}^T$. The scale factor can be recovered from A by taking its determinant:

$$s = \sqrt{\mathrm{detA}} \tag{23}$$

Taking this scale as the control error function, $e = s = \sqrt{\mathrm{detA}}$, the desired control for the ROV vertical propeller is given by:

$$\bar{\mathbf{v}}_{\mathrm{rov}}^{*} = K_p\mathbf{e} + K_d\dot{\mathbf{e}} + K_i\int \mathbf{e}dt \tag{24}$$

Where $\bar{\mathbf{v}}_{\mathrm{rov}}$ in this case is the resolved ROV vertical speed and a PID design was adopted for dynamic compensation.

E. Image stabilization with a pan- and tilt camera

The use of kinematic models for visual servoing is not always realistic for floating vehicles with relative slow dynamics. Therefore it is likely that during station keeping maneuvers, the target gets out of view due to limited bandwidth in acceleration. In an attempt to avoid these situations, an image stabilization technique is used, aiming at centering the target in the image by controlling the camera pan and tilt angles.

The pan and tilt unit, installed in the ROV, is modeled by a jacobian, that relates the angular pan and tilt velocities to the resulting camera velocity screw:

$$\mathbf{v}_{cam} = J_{pan/tilt}\mathbf{w} \tag{25}$$

Where $\mathbf{w} = [\omega_{pan}, \omega_{tilt}]^T$ contains the pan and tilt velocity components and $J_{pan/tilt}$ is in general a function of the current pan and tilt angles. For image stabilization, an image based visual servoing strategy is adopted that uses the same image error function as the station keeping controller, thus regulating the landmark centroid to the image center. Combining (25) and (15), the resolved pan and tilt velocities that guarantee exponential convergence of the image error function are given by:

$$\mathbf{w}^{*} = -\lambda\big(\mathrm{L}\mathrm{J}_{pan/tilt}\big)^{+}(s - s_d) \tag{26}$$

Where L is the image jacobian, and $\mathbf{s}$ contains the centroid coordinates.

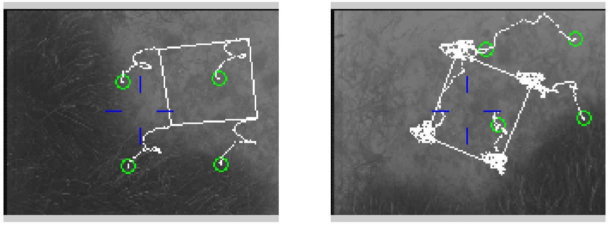
Since both the station keeping and the image stabilization controllers use the same error function, we need to decouple these tasks when simultaneously executed. This is done by transforming the station keeping error according to an homography that maps the measured image points back to a view which would have been obtained if no pan and tilt increments were applied. It follows that such a homography can be obtained from the rigid camera rotation, according to:

$$H_{pan/tilt} = KR(\theta_{pan}, \theta_{tilt})K^{-1} \tag{27}$$

Where K contains the camera intrinsic parameters. With the inverse of $H_{pan/tilt}$, it is possible to undo the deformations in the image plane due to the camera pan and tilt. To do so, a measure of the real pan and tilt angle should be available.

F. *Sea-trial results*

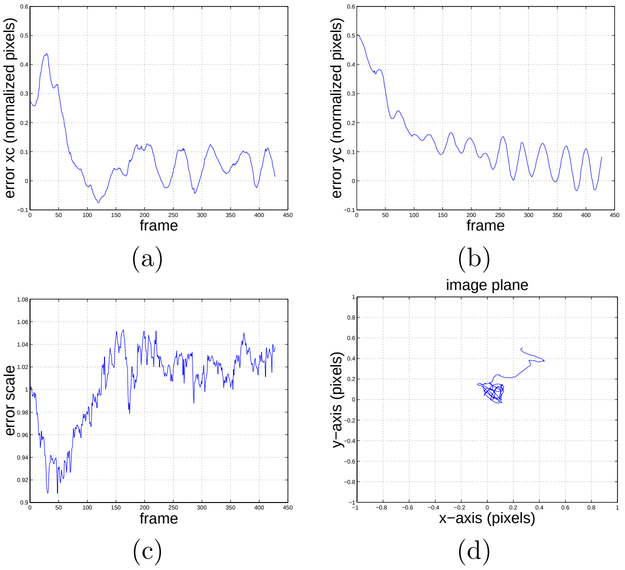
Several successful station keeping trials were performed with our ROV-system at open sea. The system was tested under various environmental conditions at different locations, namely in the North Sea near Orkney, Scotland, as well as in the Mediterranean sea in Villefranche, France. The results of a station keeping test (without image stabilization) in the Mediterranean sea are shown in Fig. (12). In a first stage, the vehicle floats uncontrolled when a landmark is selected around the image center and tracked in the presence of drift. Note that even with poor texture, the tracker was able to accurately track the selected image region. Then the visual feedback loop is closed and the landmark is driven back towards the image center, where it remains oscillating around the desired position under external disturbances. The evolution of the error signals are shown in Fig. (13) and show the convergence of the errors for the station keeping controller and the auto-depth controller.



(a) (b)

Fig. 12. Station keeping experiment at the Mediterranean: (a) tracking a selected image region in the presence of drift, with the ROV uncontrolled; (b) Controlling the centroid back to the image center by servoing the vehicle.

No efforts are made to control the landmarks orientation towards a desired value. The main difficulties arise for lateral offsets of the centroid in the image plane. In this case, since the vehicle has no lateral controllable degrees of freedom, the only solution is to compensate these errors by rotating the ROV, resulting into complex curved trajectories of the centroid and the landmark corners in the image. Such trajectories might drive the tracked region partially out of view, especially when the landmark was initially selected near to the image borders. With image stabilization, these situations can be avoided, since both lateral and frontal offsets can now be compensated by controlling the camera pan- and tilt angles. This results into trajectories that drive the landmark corners directly to the image center.



(a) (b) (c) (d)

Fig. 13. Evolution of the error signals: (a) x-coordinate of the centroid, (b) y-coordinate of the centroid, (c) relative scale, (d) centroid trajectory in the image plane.

In Fig. (14), the advantages of using image stabilization are shown. Image point trajectories are such that they drive the points directly in a straight line to their desired positions. Also the amplitude of oscillating around the desired position is kept smaller by compensating with the pan and tilt degrees of freedom.

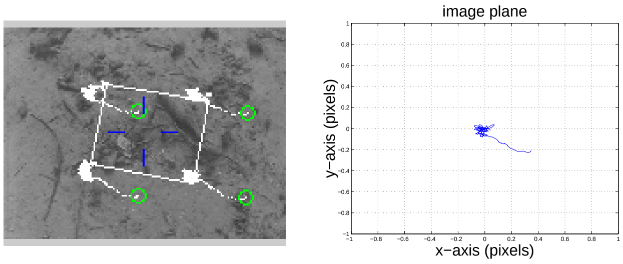


Fig. 14. Trajectory of image points during station keeping at the Mediterranean with image stabilization.

V. CONCLUSIONS

In this paper we have discussed two applications of computer vision for underwater vehicles. The motivation for using vision in the underwater scenario is driven by the high-resolution, high-bandwidth characteristics of this sensing modality. The applications can be very general and we have described here two examples: Video Mosaicking and Station Keeping.

Video mosaics can extend the navigation autonomy of camera-equipped underwater vehicles, in two main aspects: (1) By making use of non-consecutive image overlaps, it provides a precise position and motion estimation when compared with other common sensing modalities such as sonar, compass and gyroscopes. (2) It enables the creation of high accuracy mosaics that can be used as maps for posterior localization and servoing.

Station Keeping can be accomplished by tracking features in the image and by generating the appropriate closed-loop controls to maintain the vehicle stationary with respect to the sea bottom or some observed object. We have presented results obtained during sea tests.

These two applications show that vision can not only be useful to build alternative representations of the sea bottom (which can be useful in itself for mapping or visualization) but it can also be used directly in closed loop control. In the future we plan to combine these two aspects of our work and design methodologies for mosaic-based navigation, whereby a user could specify a trajectory to follow directly over the mosaic and the vehicle would later on track this trajectory in closed loop.

Acknowledgments

This work has been funded by ESPRIT-LTR project 30185, NARVAL.

References

[1] M. Gleicher. Projective registration with difference decomposition. *IEEE Conf. of Computer Vision and Pattern Recognition*, pages 331–337, jun 1997.

[2] S. J. Reeves and L. P. Heck. Selection of observations in signal reconstruction. *IEEE Trans. Signal Processing*, in press.

[3] J. F. Lots, D. M. Lane and E. Trucco. Application of 2 1/2 D visual servoing to underwater vehicle station keeping. *Proc. IEEE Oceans Conference*, Providence, USA, Sep. 2000.

[4] P. Rives and J. Borrelly. Visual servoing techniques applied to an underwater vehicle. *Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA97*, Albuquerque, New Mexico, April 1997.

[5] R. Garcia, J. Battle, X. Cufi and J. Amat. Positioning an underwater vehicle through image mosaicking. *Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA2001*, Seoul, Korea, May 2001.

[6] R. L. Marks, H. H. Wang, M. J. Lee and S. M. Rock. Automatic visual station keeping of an underwater robot. *Proc. IEEE Oceans Conference*, Brest, France, Sept. 1994.

[7] X. Xu and S. Negahdaripour. Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, (9(3)):163-184, 1992.

[8] S. van der Zwaan. Vision based station keeping and docking for floating robots. *MSc. thesis*, available at www.isr.ist.utl.pt/labs/vislab, Lisbon, May 2001.

[9] 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.

[10] S. Hutchinson, G. D. Hager and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5), 1996.

[11] J. Santos-Victor and G. Sandini. Visual behaviours for docking. *Computer Vision and Image Understanding*, 67(3):223-238, September 1997.

[12] M. Subbarao and A. Waxman. Closed form solutions to image flow equations for planar surfaces in motion. *Computer Vision Graphics and Image Processing*, 36, 1986.

[13] Thor I. Fossen. *Guidance and control of ocean vehicles.* John-Wiley & Sons, 1995.

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

[15] NARVAL homepage. http://gandalf.isr.ist.utl.pt

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

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

[18] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, September 1988.

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

[20] R. Garcia, J. Batlle, X. Cufí, and J. Amat. Positioning an underwater vehicle through image mosaicking. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 2779–2784, Seoul, Korea, May 2001.

[21] N. Gracias. Application of robust estimation to computer vision: Video mosaics and 3–D reconstruction. MSc. thesis, www.isr.ist.utl.pt/labs/vislab, Lisbon, Portugal, April 1998.

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

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

[24] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings Alvey Conference*, pages 189–192, Manchester, UK, August 1988.

[25] J. Heikkilä and Olli 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. IEEE Computer Society Press.

[26] 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.

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

[28] 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.

[29] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[30] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. ECCV.* Springer-Verlag, June 1998.

[31] B. Triggs. Autocalibration from planar scenes. In *Proc. of the European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.

[32] 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*, RA-3(4):323–344, 1987.

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

[34] J. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, October 1992.