Real-time Vision-based Station Keeping for Underwater Robots

Sjoerd van der Zwaan , José Santos-Victor

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, Torre Norte 7.26, 1049-001 Lisboa, Portugal
{sjoerd,jasv}@isr.ist.utl.pt

Abstract— ***In this paper we describe the design and implementation of a control system for automatic vision-based station keeping with an underwater ROV, relative to some visual landmark. First, a region-based tracking system is described that tracks naturally textured landmarks in the image plane, based upon full planar-projective motion models. The implementation of the algorithm is such that it allows most of the computation to be done off-line, resulting in fast and accurate tracking at near video rate. Robustness is added to the algorithm by integrating optic flow information and by learning expected image motions. Furthermore, the problem of automatically selecting promising landmarks in underwater images is addressed, based upon the information contained in the texture map. We then turn to the vision-based control problem that uses the information provided by the tracker, to station keep an underwater ROV. A decoupled control design is presented for positioning the vehicle relative to a visual landmark, while maintaining a fixed depth. Additionally, an image stabilization technique is developed, which aims to keep the landmark centered in the image plane during station keeping maneuvers, by using a pan and tilt camera. The system was tested under real conditions and results obtained from station keeping tests at open sea are presented.***

I. INTRODUCTION

Visual control loops have been introduced in order to increase the flexibility and the accuracy of underwater vehicles. In this paper, we use vision to extract information about the position and orientation of an underwater ROV, relative to some naturally textured, planar region. The work presented is integrated in the NARVAL[1] project, for which one of the main goals is the design and implementation of reliable navigation systems for mobile robots in unstructured environments. The problem addressed is that of automatic station keeping based on visual input. The station keeping task is defined locally in the neighborhood of some visually observable landmark and consists in stabilizing the vehicle relative to this landmark so as to reject 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.

Several references can be found on automatic station keeping using vision [5], [7]. In most cases the emphasis is on the controller design in a visual servoing framework, assuming the tracking of features in the image plane to be perfect [3], [4]. However, the performance of the station keeping controller strongly depends on the quality of tracking. Robust tracking of features in submarine images is hard to accomplish since in general, the images contain non-uniform lighting, low contrast, marine snow and lack of necessary features. Previous approaches that focused on accurate feature tracking in underwater sequences [6] demanded a high computational load, making the method less suitable for real-time implementation using off-the-shelf equipment.

In this paper, we describe both the tracking and the control aspects as part of a ROV-system that was successfully tested under real conditions, at open sea. A commercially available Phantom ROV is equipped with an on-board camera and adapted for computer control. A selected image region is used as a visual landmark, whose temporal changes, induced by the vehicle 's motion, are tracked through the video sequence and used for station keeping.

For tracking, we minimize the sum of squared differences (SSD) between an image region in the current view and the desired view, subject to a parameterized deformation model. We assume that the scene can be locally approximated by planar regions so that inter-image deformations are completely described by planar projective transformations. These are estimated using a set of motion models that account for expected image deformations over time. When applied to the reference image, most of the calculation can be done off-line, resulting in fast tracking, suitable for real-time implementation. To enhance robustness, the set of motion models is adapted according to the history of vehicle/camera motion, thus predicting future deformations in the image plane. We also use optic flow information to provide the tracker with an initial estimate of the current transformation parameters. Furthermore, 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.

This paper is divided in two parts. We first describe the tracking system, detailing all of the aforementioned aspects and highlighting its performance. We then turn to the control problem by describing the station keeping controller and the image stabilization technique. Station keeping results, obtained from experiments with the ROV-system at open sea, are finally presented.

[1] ESPRIT-LTR Project 30185, NARVAL - Navigation of Autonomous Robots via Active Environmental Perception, http://gandalf.isr.ist.utl.pt/narval/index.htm

II. Tracking of Planar Image Regions

Given a reference image or *template* T and a target image I, the tracking 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 in 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, $\mathcal{W}(\mathbf{q}, T)$ is the image obtained from warping T according to the transformation parameters $\mathbf{q}$.

We assume that the target to track belongs to a planar surface. For an underwater scenario, this is a reasonable assumption, since the sea bottom often can be approximated locally as a planar surface. Planar motions cannot be adequately modeled by simple image transforms, like affine or translational. A projective planar transformation is the exact motion model when a camera rotates about its focal center or if the imaged surface is planar. The 2D projective transformation is given by the 3×3 homography, H, such that $\mathbf{x}' = H\mathbf{x}$, where $\mathbf{x}'$ and $\mathbf{x}$ are the homogeneous coordinates of (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. Moreover, this homography can be decomposed into a hierarchical chain of transformations in the image plane [14]:

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

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 $|K| = 1$, 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 are used for parameterization rather then the entries of H. This parameterization is such that a zero valued parameter vector specifies the identity transform.

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:

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

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:

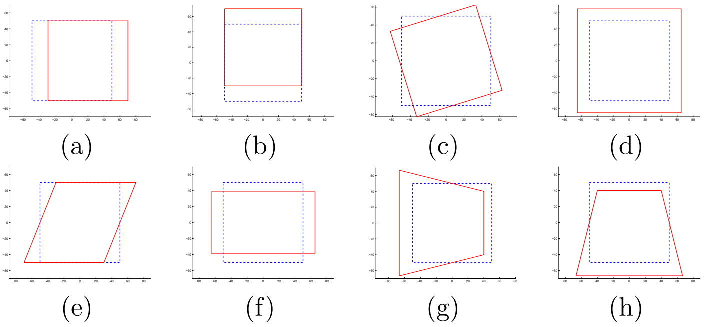


Fig. 1. Degrees of freedom of the planar projective transformation when applied to an image patch: (a) translation along the horizontal image axis, (b) translation along the vertical image axis, (c) rotation, (d) scaling, (e) shear, (f) aspect ratio, (g) projective distortion along the horizontal image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

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

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:

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

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

A. Optic Flow

In our system, we also include optic flow information in the prediction phase by assuming that the camera observes a plane and 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 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.

B. Difference Template Matching

To find the best residual warp at each time step, we minimize the error function in (3), using a set of m motion vectors $\{\boldsymbol{\Delta}\mathbf{q}_i : i \in (1 \ldots m)\}$ 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 set is constructed from sampling into the directions of the individual transform parameters, 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, $\boldsymbol{\Delta}\mathbf{q}_i$:

$$\Delta\mathbf{q} = \sum_{i=1}^{m} k_i \boldsymbol{\Delta}\mathbf{q}_i \tag{5}$$

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

$$\mathcal{W}(\mathbf{k}, T) = \mathcal{W}(\sum_{i=1}^{m} k_i \boldsymbol{\Delta}\mathbf{q}_i, T) \tag{6}$$

where $\mathcal{W}(\mathbf{k}, T)$ is the image obtained from warping the template T according to the linear combination of motion vectors $\boldsymbol{\Delta}\mathbf{q}_i$. Substituting (6) 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:

$$\mathbf{k} = \arg\min_{\mathbf{k}} \big(\frac{1}{2} \parallel \mathcal{W}^{-1}(\mathbf{q_0}, I) - \mathcal{W}(\mathbf{k}, T) \parallel^2\big) \tag{7}$$

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. In order to minimize this error function, we approximate $\mathcal{W}(\mathbf{k}, T)$ with a first order Taylor expansion, for small deviations about $\mathbf{k} = \mathbf{0}$:

$$\mathcal{W}(\mathbf{k}, T)\Big|_{\mathbf{k}=0} \approx T + \sum_{i=1}^{m} k_i \frac{\partial \mathcal{W}(\mathbf{k}, T)}{\partial k_i}\Big|_{\mathbf{k}=0}$$

where discrete approximations of each partial derivative can be expressed as:

$$\frac{\partial \mathcal{W}(\mathbf{k}, T)}{\partial k_i}\Big|_{\mathbf{k}=0} = \mathcal{W}(\mathbf{q_i}, T) - T = B_i$$

In [1], the set of vectors B_i are denoted *Difference Templates* and are also used for image registration, but they are justified in a different form. Computing each difference image, B_i, according to the motion model $\mathbf{q_i}$, and stacking them into a partial derivatives matrix: $B = [B_1 \ldots B_m]$, the image $\mathcal{W}(\mathbf{k}, T)$ can then 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 (7), a least square solution can be computed for $\mathbf{k}$:

$$\mathbf{k}_{LS} = (B^T B)^{-1} B^T D \tag{8}$$

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 $\Delta\mathbf{q}$ can be calculated from equation (5).

Most computational requirements are associated with the computation of the pseudo-inverse, $(B^T B)^{-1} B^T$, 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 to real time tracking applications.

C. Learning Motion Models

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 in the partial derivatives matrix, B, implying on-line calculation of its pseudo-inverse $(B^T B)^{-1} B^T$. 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. This is done by considering $(B^T B) = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$ as a block matrix, whose inverse is given by:

$$\begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} E^{-1} + E^{-1}FS^{-1}GE^{-1} & -E^{-1}FS^{-1} \\ -S^{-1}GE^{-1} & S^{-1} \end{bmatrix}$$

Exploiting this property, it follows [8] that the new pseudo-inverse, containing the substituted difference template, can be obtained at a negligibly extra computational effort. By on-line learning of relevant motion models and adapting the partial derivatives matrix according to them, the tracking system was able to track image deformations over a much wider range, thus adding robustness to the algorithm.

D. 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:

$$D = B\mathbf{k} + u \tag{9}$$

where u is additive noise, $\mathbf{k}$ represents the real transformation parameters that are looked for and B is the partial derivatives matrix containing all difference images. The least-square estimate for $\mathbf{k}$ is given in (8) and can be rewritten using (9) as:

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

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 $\big((B^T B)^{-1} B^T\big)u$ is minimized. The partial derivative matrix B is a function of the selected landmark texture and the set of motion models. For the same set of motion models, different landmarks result in different values of uncertainty.

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

$$\|\mathbf{k} - \mathbf{k_{LS}}\|^2 = \|\big((B^T B)^{-1} B^T\big)u\|^2 \tag{11}$$

Assuming zero-mean, unit variance white noise for u, we can take the expected value of (11), which can be computed as:

$$E\big\{\|\big((B^T B)^{-1} B^T\big)u\|^2\big\} = trace\big((B^T B)^{-1} B^T\big) \tag{12}$$

The optimal template is then found by minimizing the expected value of (12), given the set of motion models.

Fig. (2) shows the most and least informative template in an underwater image, for a fixed size landmark. These were found by performing an exhaustive search over the image space.

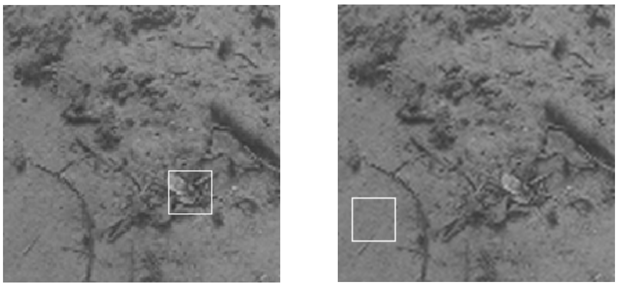

(a) (b)

Fig. 2. Automatic landmark selection: (a) most informative image region , (b) least 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 check for good environmental regions for station keeping.

E. Tracking Performance

With our tracking system, 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. (3) 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.

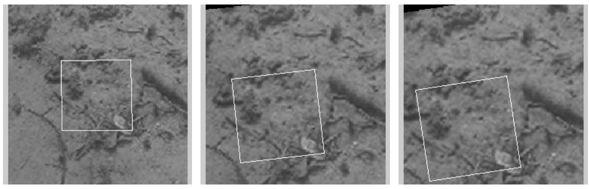


Fig. 3. Tracking an image region in a submarine video sequence.

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. (4) 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.

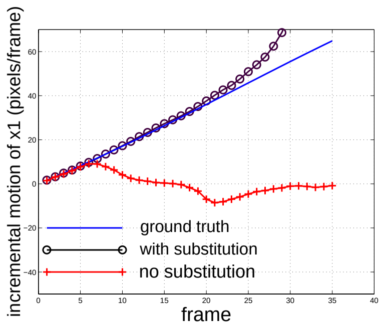
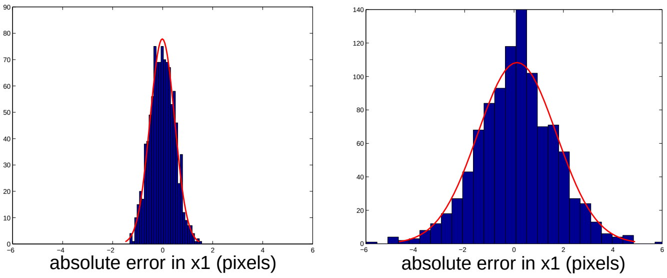


Fig. 4. Tracking the position of a corner coordinate of a selected image patch, in the presence of increasing inter-image motion.

The accuracy of the algorithm is characterized by evaluating the tracking error on images that contain randomly applied deformations with super-imposed 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 least informative templates from Fig. (2). The results are plotted in Fig. (5) and show that sub-pixel accuracy is obtained when tracking good image regions.

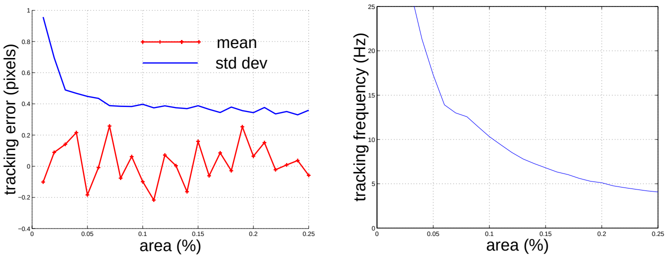
We found that the accuracy depends on the size of the tracked image region. This is illustrated in Fig (6a), where the tracking error is evaluated for different landmark sizes. Higher accuracy is obtained for larger patches, since they contain more information in the texture map. However, the



(a) (b)

Fig. 5. 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 least informative template (b).

number of pixels contained in the landmark also influences the on-line computation time necessary to run the tracking algorithm. This is illustrated in Fig. (6b). Obviously, some trade-off has to be made.



(a) (b)

Fig. 6. (a) Tracking error as a function of the landmark size (relative to the image size). For each size, the error is evaluated over a 1000 trials with images containing randomly generated deformations and added 0 mean, 10% standard deviation Gaussian noise; (b) Tracking frequency drop-off as a function of the landmark size.

III. AUTOMATIC STATION KEEPING

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 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. *ROV Description and Modeling*

A commercially available Phantom 500SP ROV is used for demonstration, which is adapted for computer control. The ROV is illustrated in Fig. (7) and is equipped, among other sensors, with an on-board pan and tilt camera. The camera is mounted rigidly to the ROV, such that its optical axis is aligned with the vertical axis of the ROV reference frame. The pan and tilt angles can be controlled separately, resulting in two extra degrees of freedom for the camera. The ROV is wired to a remote processing unit by a 150m umbilical. Video signals are sent up to the ground surface and are acquired at 25 Hz on a Matrox-board, hosted in a 450 Mhz remote computer, running Windows NT. Here, control signals are derived and sent down to the ROV via the umbilical, through a a serial communication link. This link is also used for sensor readings so that limited bandwidth is available. Therefore, the control loop is fixed at a 10 Hz frequency. The controllable degrees of freedom are defined by the geometric arrangement of the thrusters. The ROV was originally designed for joystick-type piloting, where a forward/backward force and a differential torque are commanded by two thrusters placed in the back of the vehicle and an upward/downward force is commanded through a vertically placed thruster at the vehicle center. With this arrangement, non-holonomic motion constraints are specified for the vehicle, requiring complex maneuvers for controlling the vehicle to a desired view in the image plane.

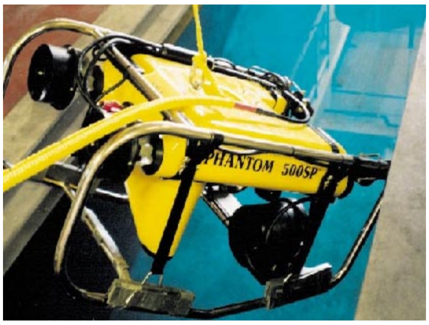


Fig. 7. Computer controlled Phantom ROV with an on-board pan- and tilt-camera.

In Fig. (8), an open-loop model is presented for the ROV [13]. The thruster commands result into generated forces and torques on the vehicle body, as given by the affine thruster model:

$$\tau = B\mathbf{u} \tag{13}$$

Where τ is the 6×1 forces and torque vector, B is the thruster model, capturing the relations of the thruster DC-motors and generated forces from the propellers and $\mathbf{u}$ contains the common mode, differential mode and vertical control inputs. The ROV dynamics is described by resolving the Newton-Euler equations of motion, which solve for the vehicle acceleration. Upon integration, the ROV instantaneous velocity, $\mathbf{v}$, is obtained and is related to the world referenced velocity, $\mathbf{n}$, via the Jacobian J.

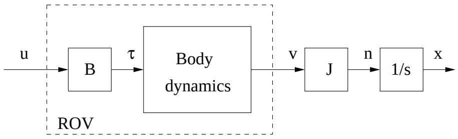


Fig. 8. Open-loop ROV model.

B. Visual Station Keeping Controller

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]:

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

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:

$$\begin{bmatrix} \dot{x_c} \\ \dot{y_c} \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x_c}{Z} & x_c y_c & -(1+x_c^2) & y_c \\ 0 & -\frac{1}{Z} & \frac{y_c}{Z} & (1+y_c^2) & -x_c y_c & -x_c \end{bmatrix} \mathbf{v}_{cam} \tag{15}$$

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

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

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:

$$\mathbf{v}_{cam} = \mathrm{J}_{rov}\bar{\mathbf{v}}_{rov} \tag{17}$$

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, $\mathrm{J}_{rov} = f\left({}^{rov}R_{cam}, P_{cam}\right)$ 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 (17) into (14), an expression is obtained that relates image point velocities to the vehicle velocity:

$$\dot{\mathbf{s}} = \mathrm{L}\mathrm{J}_{rov}\bar{\mathbf{v}}_{rov} \tag{18}$$

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

$$\bar{\mathbf{v}}^*_{rov} = -\lambda\big(\mathrm{L}(\mathbf{s}, Z)\mathrm{J}_{rov}\big)^{+}(s - s_d) \tag{19}$$

This expression takes the vehicle motion constraints into account, resulting into trajectories that are physically executable. In Fig. (9), the overall control system design is given for station keeping in the horizontal plane. The desired controls for the left and right back thrusters, $\mathbf{u}_{\mathrm{h}} = [\mathrm{u_l}, \mathrm{u_r}]^T$, can be calculated from combining the relevant entries of the thruster model in (13) with the kinematic control law in (19):

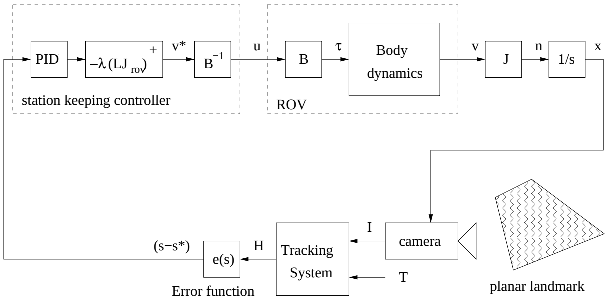


Fig. 9. Visual control loop for station keeping with the ROV.

$$\mathbf{u}_{\mathrm{h}} = -\lambda \mathrm{B}_h^{-1}\big(\mathrm{L}(\mathbf{s}, Z)\mathrm{J}_{rov}\big)^{+}(K_p\mathbf{e} + K_d\dot{\mathbf{e}} + K_i\int \mathbf{e}dt) \tag{20}$$

Here we have included a PID control action on the image error for dynamic compensation.

C. 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:

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & \lambda \end{bmatrix} \tag{21}$$

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:

$$s = \sqrt{|\mathrm{A}|} \tag{22}$$

Taking this scale as the control error function, the desired control for the ROV vertical propeller is given by:

$$\mathbf{u}_{\mathrm{v}} = -\mathrm{B}_v^{-1}(K_p\mathbf{e} + K_d\dot{\mathbf{e}} + K_i\int \mathbf{e}dt) \tag{23}$$

Where B_v is the relevant entry of the thruster model in (13), corresponding to the vertical thruster and a PID design was adopted for dynamic compensation. The overall vertical control system design can be represented by Fig. (9) by simply substituting the task function , $\mathbf{e}(\mathbf{s})$, and the controller block.

D. Image Stabilization

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:

$$\mathbf{v}_{cam} = J_{pan/tilt}\mathbf{w} \tag{24}$$

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 (24) and (14), the resolved pan and tilt velocities that guarantee exponential convergence of the image error function are given by:

$$\mathbf{w}^* = -\lambda\big(\mathrm{L}\mathrm{J}_{pan/tilt}\big)^+(s - s_d) \tag{25}$$

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:

$$H_{pan/tilt} = KR(\theta_{pan}, \theta_{tilt})K^{-1} \tag{26}$$

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 panning and tilting of the camera. To do so, a measure of the real pan and tilt angle is necessary.

The overall control design for the station keeping controller with image stabilization is illustrated in Fig. (10). The same modifications apply for the visual auto-depth controller.

E. 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 in the Mediterranean sea are shown in Fig. (11). In a first stage, the vehicle floats uncontrolled when a landmark is selected around the image center and tracked in the presence of drift. Even with poor texture, the tracker was able to accurately track the selected image region. Upon closing the visual feedback loop, 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. (12) and show the convergence of the centroid and the scale of the landmark.

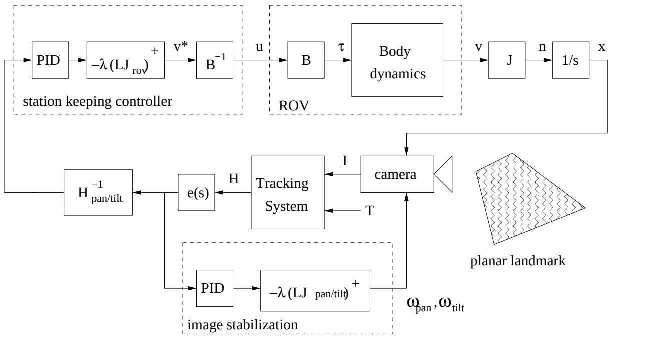
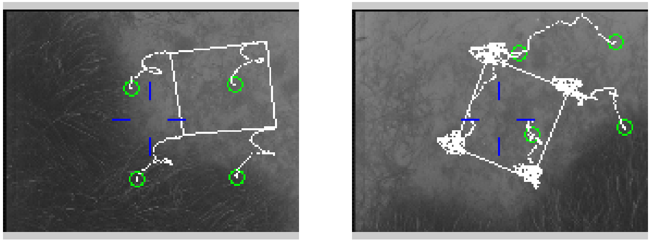


Fig. 10. Visual control loop for station keeping with image stabilization.

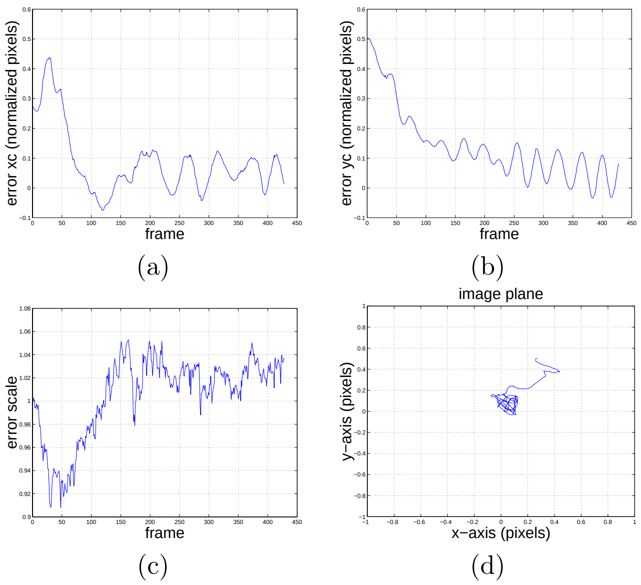


(a) (b)

Fig. 11. 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 so as 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 plane. Such trajectories might drive the tracked region partially out of view, especially when the landmark is 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.

In Fig. (13), 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 and the amplitude of oscillating around the desired position is kept smaller.



(a) (b)

(c) (d)

Fig. 12. Evolution of the error signals during a station keeping maneuver: (a) x-coordinate of the centroid, (b) y-coordinate of the centroid, (c) relative scale, (d) centorid trajectory in the image plane.

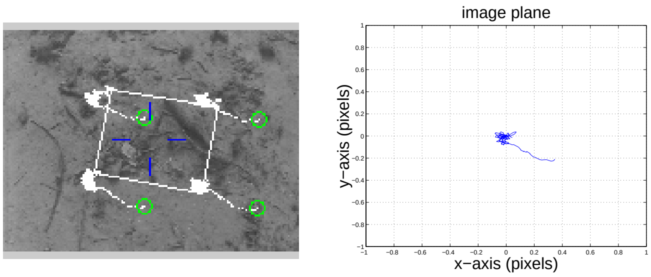


Fig. 13. Trajectory of image points during station keeping at the Mediterranean with image stabilization.

IV. Conclusions

In this paper we presented the tracking and control aspects for automatic visual station keeping with an underwater ROV. Tracking of image regions was realized by integrating optic flow information with a correlation based optimization method, resulting in sub-pixels accuracy. Planar projective motion models were considered that cover the whole range of image deformations that occur when a camera moves in 3D. For correlation, a set of motion models was used, sampling for expected image deformations. The main advantage is that these can be pre-calculated when applied to the desired view, resulting in high tracking frequencies. To enhance robustness, the set of models was iteratively adapted to the history of detected camera motion. Also, the importance of landmark selection w.r.t tracking accuracy was shown and a method was described for automatically selecting the most informative image region.

Using the tracker information, visual control loops were designed to perform station keeping. The station keeping task was formulated in the image plane and a decoupled control strategy was adopted. For station keeping, we considered the regulation of the landmark centroid towards the image center, while not controlling its orientation towards a final value at all. The main motivation was that, given the vehicle motion constraints, lateral offset in the image plane can only be compensated by rotating the ROV.

With the use of an image stabilization technique, the overall system gained more robustness. The advantages were twofold: (i) it prevents the landmark of going out of view and (ii) observed image transformations are kept small. To decouple image stabilization from station keeping, error signals needed to be transformed, requiring the measurement of the camera pan and tilt angles as well as the camera intrinsic parameters.

This system was successfully tested under real conditions at open-sea. For future work, we consider to include the vehicle dynamics into the tracking system and the controller design.

Acknowledgments

This work has been funded by ESPIT-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 Transactions on 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. of the IEEE Oceans Conference*, Providence, USA, September 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. of the IEEE Oceans Conference*, Brest, France, September 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 http://www.isr.ist.utl.pt/labs/vislab/thesis/, 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.