Computer Vision and Image Understanding **79,** 47–65 (2000)
doi:10.1006/cviu.2000.0849, available online at http://www.idealibrary.com on IDEAL®

Dynamic Stabilization of a Pan and Tilt Camera for Submarine Image Visualization

Armel Crétual[1] and François Chaumette[2]

IRISA/INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France
E-mail: armel@ltsi.univ-rennes1.fr, chaumett@irisa.fr

Received March 12, 1999; accepted February 4, 2000

One major problem of underwater observation with an automatic engine is the instability of image acquisition and visualization. Indeed, small engines of this kind are subjected to low-frequency motions due to weak friction and water currents. In this paper, we propose to maintain stabilization in the image by controlling the pan and tilt motions of the camera attached to the engine, using techniques for target tracking by visual servoing. The main idea behind approach lies in the fact that, since it is very difficult to track a point in the images of an unknown and complex scene using geometrical tools, the position of a virtual point can be retrieved by the integration of its 2D motion. The motion estimation method we have used, called the RMR algorithm, provides the parameters of a selected motion model (for the task considered here, a constant one) and is perfectly suitable for real-time constraints and the complexity of an undersea image sequence. Our approach has been validated on a dry setup using two different sequences of underwater images.

Key Words: image stabilization; visual servoing; 2D motion; underwater vision.

1. INTRODUCTION

The automatic exploration of the seafloor is a challenging application for the combination of vision and robotics. Until now, the great majority of submarine engines dedicated to observation have still been, at least partially, manually guided. We are interested in this paper in the automatic stabilization of the image sequence acquired by a pan and tilt camera mounted on a remote operated vehicle (ROV). The topic of image stabilization is indeed important in underwater vision for applications relying on fixed, or slowly moving, image acquisition, such as those requiring human remote observation. Two examples are

[1] Current address: Laboratoire du Traitement du Signal et de l'Image (LTSI). Université de Rennes I, Campus de Beaulieu, 35042 Rennes Cedex, France.

[2] To whom correspondence should be addressed.

1077-3142/00 $35.00

observation of active hydrothermal vents and biological surveys of interacting animals. Even if the ROV is supposed to be motionless, it is well known that due, first, to the weak friction between the vehicle and the water and, then, to undertows, the ROV is subject to undesirable motions. The image stabilization will be performed by a control of the pan and tilt degrees of freedom (d.o.f.) of the camera. Obviously, controlling only these two d.o.f. does not allow us to compensate for every motion in the image. All the complex deformations, meaning, for example, zoom effects or rotations around the optical axis, will remain, even if the stabilization is performed. However, if we consider classical problems of underwater image analysis such as fish numbering or pipeline state control, the pan and tilt stabilization is sufficient, and there is no need to offset the other deformations.

The pan and tilt control we want to rely on is also exploited in many tracking tasks using a mobile camera. The link is easy to explain since, as seen from the observer point of view (here, the camera), a motion of an object in the image can be due to the actual motion of the object, the motion of the camera, or a combination of these two motions.

Many different studies have been made in tracking a moving target with an active camera. First, research dealing with the robotics aspect is generally not concerned with vision problems, but with contorl strategy. As a consequence, there is often very strong a priori knowledge of the observed object, in order to validate the control law. Most of these works [1–4] use a quasi-binary image to easily separate the target from the background.

On the other hand, several works have emphasized the visual processing problem of recovering the target center of gravity while using trusty methods to control this estimated position. For example, methods used in [5, 6] only allow tracking of a small object, or at best, an object which covers a much smaller part of the image than the background covers. A 2D affine motion model is computed between two successive images, and the second image is compensated with the opposite motion. Thresholding the difference between this reconstructed image and the original one gives the position of the object. The idea in [7] is quite the same except that compensation is based on the measured motion of the camera, obtained using the odometry of the robot, and its resulting 2D motion in the image. Larger objects can be tracked but there is a great limitation due to the necessary high-quality calibration of the system. In [8], a corner detection algorithm yields the position of a particular point of the object. This raises two major problems. First, ensuring stable visual tracking of points of interest has a computational cost incompatible with the robotic constraints. There is also the problem of the observability of the point of interest, which must always be seen. Finally, in [9–12] a sterovision system is used to build a 3D model of the object motion. Using 3D reconstruction once again requires good calibration of the whole system, the camera, and the robot system.

If we consider that the aim of stabilization is to maintain fixation on a scene (meaning not only an object within the field of view), there is a slight difference between tracking and stabilization. Tracking is generally applied to an object of interest which has to be centered in the image, whereas stabilization is concerned with the whole image, and the point to be maintained at the center is the one initially in that position. Thus, there is no a priori knowledge of the point of interest shape. It could even appear in an area of the image without any contrast and be impossible to discriminate from its neighbors. This uncertainty of the position would cause oscillation around the desired position of the ROV, quite undesirable. Since, in fact, only a perturbation rejection has to be performed, some studies have been made using inertial information on the camera. Indeed, if the scene is static, stabilization means that there is no motion of the camera. Thus, if such motion occurs due to a displacement of

the system on which the camera is mounted, the camera motions can be directly compensated for using the vestibular information. One of the main arguments for this approach is that, theoretically, the vestibular information is sufficient to perform the perturbation rejection. Nevertheless, as this information is the acceleration around each axis of the camera, a drift can appear in the position of the area of interest in the image. More or less, using the image information allows us to close the loop on this level and to compensate for potential deviations in the image. Thus, lower quality image processing is needed, since it only comes in addition to the vestibular compensation. This approach has been applied in [13]. Nevertheless, in this article, a binocular head was used, and only the vergence was controlled.

Recently, it was shown that visual servoing based on dynamic measurements [14–16] can be exploited in real-time applications. The main interest is that estimation of 2D-motion models does not require any visual marks, but only a texture contrasting sufficiently to reliably measure spatiotemporal gradients of the image intensities. Another point is that a better quality of motion estimation can be reached without having to increase the duration of processing greatly between two images, incontrast to points-tracking algorithms. As an example, if we consider the case of a great displacement, a motion estimation based on spatiotemporal gradients of image intensity will only need to have a coarser level within a hierarchical scheme, whereas a point-stracking algorithm will need to increase the size of the search area.

In [17–19], image sequence stabilization has been performed using 2D motion estimation, but the image processing was performed offline. This means that a whole sequence is acquired first and then new images are generated by inversion of the deformation between two successive original images. In that case, time processing is of course not critical. In [20], the stabilization is performed online, but necessitates a coarse knowledge of the relative positions of the target and the camera and the learning of eight images of the scene regularly spaced around the point of interest.

Two approaches to realizing the image stabilization can be considered. The first consists of regulating the observed motion to zero while the second consists of maintaining the area around the center of the image at the same position. The first approach raises the problem of increasing the derivation order by one compared to the second approach. This may lead to complex and quite unstable control laws. Therefore, we propose in this article to retrieve position of the point of interest (here the center of the first image) by integrating its estimated speed [21]. Thus, we can apply classical control laws, designed for geometric measurements, in order to keep the same scene in the camera field of view. A drift may also appear when this approach is used, in the case of strong accelerations which disturb motion estimation. However, such accelerations rarely occur in an undersea environment. Finally, interesting stabilization results are described in [22]. Even if it also uses the idea of integrating speed to estimate the position, this approach is different from ours since the estimation is performed in 3D space, and the aim is to retrieve the 3D position of the ROV.

The rest of the paper is organized as follows. In Section 2, we describe how the position of the point of interest can be estimated. We also briefly recall the principle of the motion estimation algorithm we have used. We detail in Section 3 the whole stabilization task from the robotics point of view, meaning which control law is used. At this stage of our study, we are still in the validation step, meaning that our tests are performed with a dry setup. Our experimental bench is presented in Section 4. Then, in Section 5, we report experimental results related to two different sequences of submarine images. Finally, a discussion and concluding remarks are given in Section 6.

2. ESTIMATION OF THE 2D POSITION OF THE INTEREST POINT

Let us denote as $s = (x, y)^T$ the 2D projection at time t of a 3D point M, and as $\dot{s}$ its apparent speed in the image. s can obviously be recovered when the projection position s_0 at time 0 and the evolution of $\dot{s}$ over time are known, by

$$\begin{aligned} s &= s_0 + \int_0^t \dot{s}\, dt \quad \text{(in continuous form)} \\ s &= s_0 + \sum_{i=1}^{k} \dot{s}_i \delta t_i \quad \text{(in discrete form)}, \end{aligned} \tag{1}$$

with $\dot{s}_i$ being the ith measurement of $\dot{s}$ and δt_i the time duration between the $(i-1)$th and ith measurements, provided by the computer clock.

The motion model used to approximate speed in the image can be, for example, a quadratic one with eight parameters as in (see [14, 23])

$$\begin{cases} \dot{x} = c_1 + a_1 x + a_2 y + q_1 x^2 + q_2 xy \\ \dot{y} = c_2 + a_3 x + a_4 y + q_3 y^2 + q_4 xy \end{cases} \tag{2}$$

$$\text{with} \begin{cases} c_1 = -\frac{T_x}{Z_p} - \Omega_y & c_2 = -\frac{T_y}{Z_p} + \Omega_x \\ a_1 = \gamma_1 \frac{T_x}{Z_p} + \frac{T_z}{Z_p} & a_2 = \gamma_2 \frac{T_x}{Z_p} + \Omega_z \\ a_3 = \gamma_1 \frac{T_y}{Z_p} - \Omega_z & a_4 = \gamma_2 \frac{T_y}{Z_p} + \frac{T_z}{Z_p} \\ q_1 = q_4 = -\gamma_1 \frac{T_z}{Z_p} - \Omega_y & q_2 = q_3 = -\gamma_2 \frac{T_z}{Z_p} + \Omega_x, \end{cases}$$

where $T = (T_x, T_y, T_z)$ and $\Omega = (\Omega_x, \Omega_y, \Omega_z)$ respectively represent the translational and the rotational components of the kinematic screw associated with the relative rigid motion between the camera frame and the object frame, $Z = Z_p + \gamma_1 X + \gamma_2 Y$ being the equation of the planar approximation of the object surface around the considered point, expressed in the camera frame.

Of course, other motion models, e.g., constant (the restriction of the presented model to the terms c_i) or affine (the restriction to the terms c_i and a_i), could be used to estimate the position of the image center. In fact, compromise between accuracy of estimation and computation load is necessary since, due to robotics constraint, the control rate must be the highest possible, that is, the closest possible to the video rate, in order to provide smooth behavior of the stabilization.

The apparent motion in the image is generally complex. In the best case, when the observed scene is an undeformable planar surface, the displacement field of each pixel of the image corresponds to the 2D quadratic transformation presented above (see for example [23] for a proof of this assertion). In any other case, the quadratic model is only an approximation of the optical flow field. When dynamic visual measurements are embedded in a real-time closed loop control scheme, a very strong constraint concerning the computational time appears. Actually, it is often difficult to estimate 2D motion models more elaborated than constant or affine ones.

Moreover, if we consider any polynomial model of motion, the speed of the image center is equal to the 0th order approximation of the image motion, meaning $(c_1, c_2)^T$. As the aim

of the task is to keep the point of interest at the image center, its two coordinates x and y will be always close to 0 if the control law is reactive enough. In that case, the precision brought by the quadratic or affine part of the model is weak compared to that brought by the constant model. Consequently, we chose to consider only the constant model to estimate the speed of our point of interest. This choice, validated by the experimental results presented in this paper, gives us an answer in the compromise between the precision of the estimation and its necessary swiftness.

However, we have to note that a limitation of using the constant parameters arises when a large part of the image has a very weak texture. As an example, in the case of pure divergent image motion, if no spatiotemporal gradient can be observed in the lower half of the image, the c_i parameters, which should be zero, will be estimated as a translation along the $\boldsymbol{y}$ axis. The only solution to the problem would be to consider a more complex motion model. As already stated, this is unfortunately impossible due to real-time constraints.

Motion parameters are computed using the robust multiresolution estimation method (RMR algorithm) described in [24]. A Gaussian image pyramid is constructed at each instant. Let Θ be the vector of the motion model parameters at instant t. On the coarsest level, the first estimation of Θ consists of minimizing with respect to Θ the criterion

$$C(\Theta) = \sum_p \psi(r(p, \Theta)) \quad \text{with } r(p, \Theta) = \nabla I(p, t) \cdot w_\Theta(p) + I_t(p, t),$$

where points p are all the points of the image, I is the intensity function, ∇I and I_t are its spatial gradient and temporal derivative, $w_\Theta(p)$ is the velocity vector at point p provided by Θ, and ψ is a robust estimator, classically Tukey's biweight function. This estimator allows us to reject the outliers, i.e., points p whose spatiotemporal gradient does not correspond to the current estimation of Θ.

Then, a hierarchical and iterative strategy is used. Let Θ_i^k be the estimate of Θ at level i and kth iteration at this level. We have $\Theta_i^k = \Theta + \Delta\Theta_i^k$. Successive incremental refinements $\Delta\Theta_i^k$ are given by

$$\Delta\Theta_i^k = \arg\min_{\Delta\Theta_i^k} \sum_p \psi\left(r'\left(\Delta\Theta_i^k\right)\right),$$

with $r'(\Delta\Theta_i^k) = \nabla I(p + w_{\Theta_i^k}(p), t+1) \cdot w_{\Delta\Theta_i^k}(p) + I(p + w_{\Theta_i^k}(p), t+1) - I(p, t)$.

Once the iterative estimation Θ_i of Θ is performed at level i using the incremental optimization process, the estimation at level $i+1$ is initialized by the projection of Θ_i on this finer resolution level, and this hierarchical scheme is driven up until the finest resolution level, i.e., the two initial images, is reached.

3. CONTROL LAW

Once the current position of the image center is estimated from (1), we can resort to a standard control law to realize the regulation of this estimated 2D position. The control scheme we use is the classical image-based visual servoing, presented for example in [25], which is known to be robust with respect to calibration errors.

The desired position s^* of $s = (x, y)^T$ being the image center ($s^* = (0, 0)^T$), s can be directly viewed as the vector of error. The visual servoing goal is then to bring and maintain this error to zero by controlling the camera pan and tilt. To design the control law, we use

the relation between the temporal variation of s and the camera motion. As this motion is restricted to rotational velocities $\Omega_{c,x}$ (pan) and $\Omega_{c,y}$ (tilt), we get from (2)

$$\dot{s} = L \begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t} \quad \text{with } L = \begin{bmatrix} xy & (-1-x^2) \\ (1+y^2) & -xy \end{bmatrix},$$

where $\frac{\partial s}{\partial t}$ represents the variations of s due to other motions. In our case, the observed scene is generally a part of the motionless sea floor. The term $\frac{\partial s}{\partial t}$ thus comes from the uncontrolled motions of the ROV.

Let us note that, obviously, even if the observed scene is moving (for example, if the aim is to maintain fixation on another submarine in the field of view or to track a fish), the control law is still valid. There is only a practical potential problem in this case: it is necessary to have information on the location of the object of interest in the image—for example, that the object takes up more than 50% of the image. Thus, due to the rejection of nondominant motion provided by the RMR algorithm, estimation over the whole image would give the motion of the object. If it is not the case, or if we do not know about this, we should, at least, have the initial location of the object to track, just as is done in [26], in order to perform the estimation mainly on pixels belonging to it.

Specifying exponential decay of the error with gain λ ($\dot{s} = -\lambda s$), the control law is given by

$$\begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} = -\lambda L^{-1} s - L^{-1} \widehat{\frac{\partial s}{\partial t}}. \tag{3}$$

More precisely, the chosen gain λ is not constant but adaptive. This means that this gain is automatically tuned in order to preserve the stability of the system when the error s is great, and, in the same time, to optimize the time to convergence by increasing when the error is weak.

The first term of this control law allows the system to reach convergence. The second term has to be added in order to compensate for the perturbing motions.

The estimate $\widehat{\frac{\partial s}{\partial t}}$ of $\frac{\partial s}{\partial t}$ can be made as explained in [27] by

$$\widehat{\frac{\partial s}{\partial t}} = \hat{\dot{s}} - L \widehat{\Omega_c}. \tag{4}$$

In our case, the measure $\hat{\dot{s}}$ of $\dot{s}$ is directly supplied by (2) and the motion parameters are provided by the estimation algorithm. $\widehat{\Omega_c}$ is the measured camera rotational velocity.

If the object is motionless, a first-order approximation gives

$$s_{(k)/(k-1)} = s_{k-1} + \dot{s}\Delta t = s_{k-1} + L \widehat{\Omega_c} \Delta t, \tag{5}$$

where Δt is the sampling period. Thus, the estimate given by (4) can be seen at iteration k as

$$\left(\widehat{\frac{\partial s}{\partial t}} \right)_{(k)} = \frac{s_{(k)} - s_{(k)/(k-1)}}{\Delta t}. \tag{6}$$

This represents the discrepancy between the actual measure of the visual features and the predicted one. Let this value be null if the speed of the scene relative to the camera is null and constant if this velocity is also constant. This means that the discrepancy and the speed of the ROV relative to the sea floor have the same model.

Let us now come back to the control point of view and let us consider robustness issues. Two different sources for noise are possible in our estimation scheme: it can be introduced either through the extraction of the visual data or due to measurement errors in the camera pan–tilt velocities. To deal with this problem, we have implemented a Kalman augmented filter (see [28] for any details), with a constant acceleration state model, the equations of which are given by

$$\begin{cases} \left(\frac{\partial s}{\partial t}\right)_{(k+1)} = \left(\frac{\partial s}{\partial t}\right)_{(k)} + \Delta t \left(\frac{\partial \dot{s}}{\partial t}\right)_{(k)} + \nu_{(k)} \\ \nu_{(k+1)} = \rho \nu_{(k)} + v_{1(k)} \\ \left(\frac{\partial \dot{s}}{\partial t}\right)_{(k+1)} = \left(\frac{\partial \dot{s}}{\partial t}\right)_{(k)} + v_{2(k)}, \end{cases} \tag{7}$$

when ν expresses the correlation between the accelerations over time, ρ is the degree of correlation between two successive values (ρ can range from 0 to 1 and has been set to 0.3 in the experiments described below), and v_1 and v_2 are the zero-mean Gaussian white noises in the chosen model. Furthermore, the relation involved in the Kalman filter relating the observed data to the chosen model is given by

$$\left(\frac{\widehat{\partial s}}{\partial t}\right)_{(k)} = \left(\frac{\partial s}{\partial t}\right)_{(k)} + \omega_{(k)}, \tag{8}$$

where $\frac{\widehat{\partial s}}{\partial t}$ is given by (4), and ω is a zero-mean Gaussian white noise in the observations.

Finally, let us note that the control law given by (3) is insufficient to compensate for possible tracking errors due to nonzero target accelerations. To overcome this problem, the prediction of the target motion, provided by the Kalman filter, can be used. This leads to the following adaptive predictive control law:

$$\begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} = -\lambda L^{-1} s - L^{-1} \left(\frac{\widehat{\partial s}}{\partial t}\right)_{(k+1)/(k)}. \tag{9}$$

Furthermore, a prediction of the position s of the visual features in the image is given by

$$s_{(k+1)/(k)} = s_k + \dot{s}\Delta t = s_k + L\widehat{\Omega_c}\Delta t + \left(\frac{\widehat{\partial s}}{\partial t}\right)_{(k+1)/(k)}.$$

Its benefit comes from the fact that, in case of mismeasurement, the predicted position in the image enables us to pursue tracking by ensuring that it is a sufficiently good estimate.

4. DESCRIPTION OF THE EXPERIMENTAL SYSTEM

Before integrating our system in a ROV, it was necessary to validate our approach. The tracking task has been previously validated in [26], even with some complex objects such as a pedestrian. Nevertheless, one of the main problems in working with submarine images is the poor quality of the images. Therefore, to ratify our approach and to ensure the most realistic simulation of having a controlled camera mounted in a ROV, it was necessary to use submarine images. We present now our validation bench.

FIG. 1. Validation setup.

4.1. Validation Dry Setup for the Control Scheme

The principle of our validation bench is to use a pan and tilt camera that observes a screen on which are projected previously acquired submarine images (see Fig. 1). The objective is to observe a small part of this screen and to try to maintain the fixation on that part.

The camera is weakly calibrated (principal point at the image center, focal length, and pixel size given by the camera constructor) since the control scheme is not sensitive to calibration errors. The Sony camera we used is a CCD one. The system is driven by an UltraSparc I Sun station at 200 MHz, communicating with a Sun Video frame grabber and an AICO/ITMI numerical control unit. The image size is 128×128 pixels, even if only the central part (100×100 pixels) is used for the motion estimation. Once again, the choice of size is motivated by the necessary compromise between precision and execution speed. After several tests, this size appeared give to very acceptable results while allowing us to work at a rate of one image every 80 ms, which means a 12.5-Hz rate.

We have conditions close to reality, since we work on submarine images and with perturbations in the acquisition due to ROV motions induced by water currents. Moreover, observing a projection screen raises the problem of loss of image quality. More precisely, the submarine images have been projected onto a classical TV screen (56-cm diagonal dimension, 4-MHz bandwidth, which means medium market range equipment) using a classical video player. The screen was 1 m from the camera. The initial position of the camera at the beginning of the task is such that its optical axis is coarsely perpendicular to the plane of the screen (with precision not better than 15°). The camera is pointing near the center of the screen, to ensure a sufficient possibility of moving by keeping the screen in the field of view. The area of interest is chosen only to be easily recognizable, for the purpose of clarity in the presentation of results. No conditions on its shape or pattern are taken into account. In the best case, when the image plane of the camera is parallel to

the TV screen, the projection from the TV screen to the image plane is a pure zoom scale factor. In any other case, the projection will induce a more complex deformation, which means the image stabilization using our dry set-up is at least as difficult as that under real conditions.

Another difference between our dry setup and the reality is that, on our test bench, the platform on which the camera is mounted is fully linked to the ground. An interesting problem related to the automation field arises in reality: the influence of the camera rotations on the ROV trim. Due to the weak friction between the engine and the water, a displacement of the camera will produce an induced motion of the submarine. This is similar to the case of an astronaut in weightlessness conditions. In the case of the ROV, the motion of the camera produces a displacement of the center of mass of the whole system, which induces a small force, not absorbed by the friction. Nevertheless, the weak ratio between the camera and ROV weights and the limited amplitude of camera motions allow us to consider this effect negligible. Another reason for the same conclusion is that the perturbing motions have a low frequency, and therefore the acceleration needed for the camera is weak.

5. RESULTS

Our control scheme has been tested on two different image sequences. In each case, the motion of the submarine engine is such that the central area of the image observed at the beginning can be viewed all through the sequence. This means that the sequence must be such that this part of the image corresponds to the same pattern of the sea floor for at least 20 s, in order to test the stability of our control scheme and, more precisely, its behavior faced with possible drift due to integration of the speed. This was the only criterion for the choice of sequences in the database provided to us by Ifremer. Of course, in this database, no control was used during the image acquisition to stabilize the shooting. Furthermore, no data is available concerning the 3D motion of the ROV.

The two different sequences chosen for our experiments are respectively called “rocks” and “sharks.” In each of them, several images and curves are displayed. These are

- The reference sequence. These are the full-screen images taken by our camera. To shoot them, the camera was motionless and the focal length was changed to observe the entire TV screen. One can notice some problems due to the sweep on the TV screen, in particular because there is no synchronization between the visualization frequency on the screen and the shooting frequency of the CCD camera. On the first image of the sequences, a white square indicates the area of interest which has to be maintained motionless when the control loop is switched on. One image out of 25 is displayed, which represents approximately one every 2 s.
- The displacement in pixels of the area of interest in the image when no control is performed to compensate the ROV motion influence. The displacement along the x axis of the image (the horizontal one) is shown by a solid line, and the one along the y axis (vertical) is shown by a dotted line.
- The result sequence, meaning the part of the image which is acquired and on which the image processing is done. The size of these images relative to the screen is about 40% in height and 30% in width, which is 12% of the whole screen surface. One image out of 10 is displayed, which represents a bit more than one every second.
- Three curves of parameters acquired during the stabilization, representing.

—The displacement of the point initially at the center of the 128×128 treated image, expressed in pixels. The displacement along the x axis of the image is shown by a solid line, and the one along the y axis is shown by a dotted line.

—The measured speed of the point of interest provided by the motion estimation algorithm, expressed in pixels/s. Once again, the value along the horizontal axis x is shown by a solid line and the one along the vertical axis y is shown by a dotted line.

—The rotational controlled speeds. These are the velocities computed by the control law and given as orders to the pan and tilt platform, expressed in degrees per second. The value of the rotational speed around the x axis of the camera is shown by a solid line and the value of the rotational speed around the y axis is shown by a dotted line. Theoretically, the speed around the x axis of the camera (respectively the y axis) allows us to compensate for motions along the y axis of the image (respectively the x axis). This is true only when the robot wrist on which the camera is mounted is perfectly well calibrated. In practice, a rotation around the x axis of the camera also causes a displacement in the x direction of the image, and vice versa, providing a new source of possible errors.

5.1. "Rocks" Sequence Results

We can notice in the "rocks" sequence, presented in Fig. 2, that the area of interest undergoes a large motion in the image (and is well tracked and centered as shown in Fig. 3).

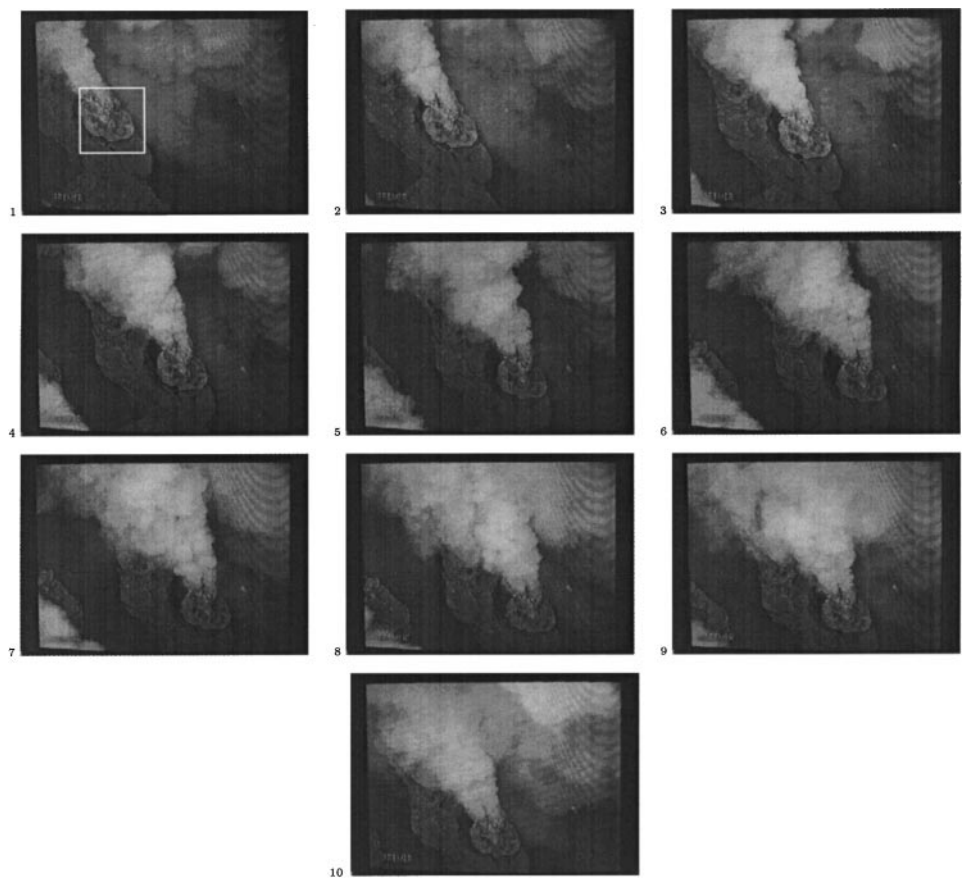

FIG. 2. "Rocks" sequence. One full image out of 25 from the original sequence (one image every 2 s) with a motionless camera.

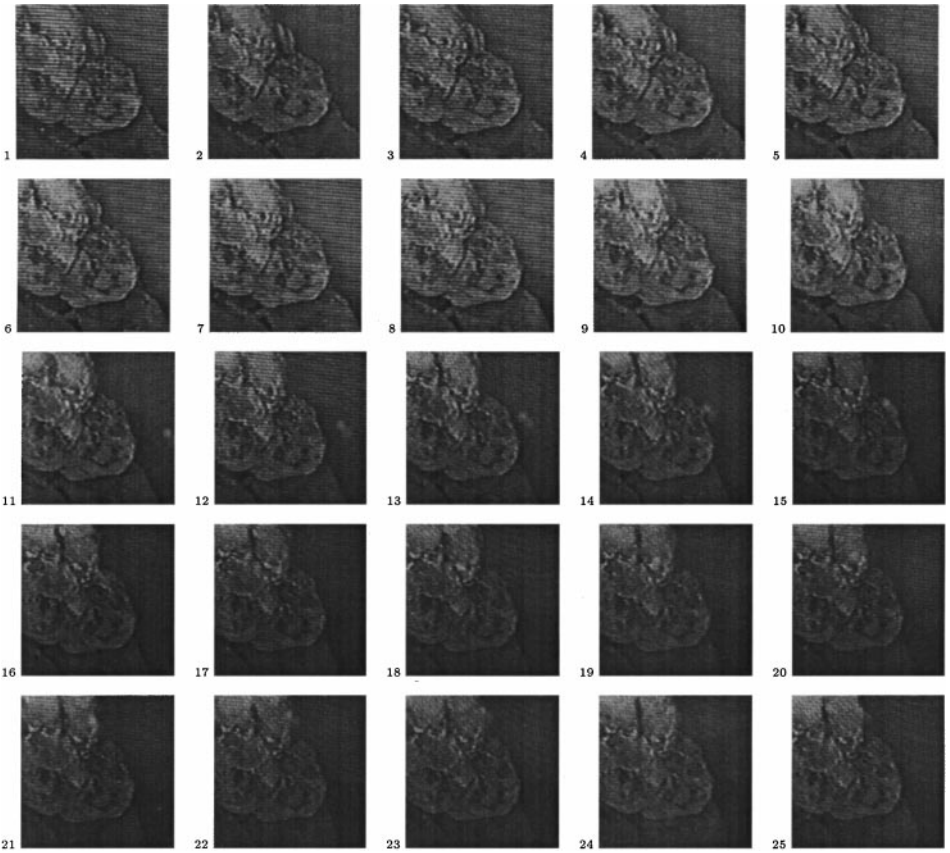

FIG. 3. One image out of 10 from the "rocks" sequence acquired during stabilization (one image every 800 ms).

This zone, the head of the rock from which smoke and gas escape, undergoes first a diagonal motion from the left of the image to the bottom right and then a motion on the screen border toward the bottom right. This is confirmed by the displacement of the area of interest when no control is performed to rotate the camera, shown in Fig. 4. We notice that the amplitude of the motion is more than 200 pixels in the x direction and more than 100 pixels in the y direction. And, for example, during the first 50 iterations, the speed in the image is approximately 20 pixels/s along the x axis and 10 pixels/s along the y axis.

The curves displayed in Fig. 5 represent the estimated position of the center of the area of interest during stabilization. As the initial position is the desired one, the aim of the task is a perturbation rejection. Thus, to evaluate the accuracy of our scheme, the main thing to look at is not the behavior of this curve, but the maximal error. It appears that the estimated position of the pixel initially at the center of our area of interest is never more than half a pixel away from the image center, which is much less than the displacement obtained when no control is performed (more than 100 pixels in each direction after 10 s). This validates at the same time the control and the estimation parts of our algorithm.

The estimated speed of the point of interest appears quite noisy in Fig. 6. This can be simply explained by the fact that it is the combination of the motion of area of interest in the image and the motion of the CCD camera. Nevertheless, this speed always remains close to zero.

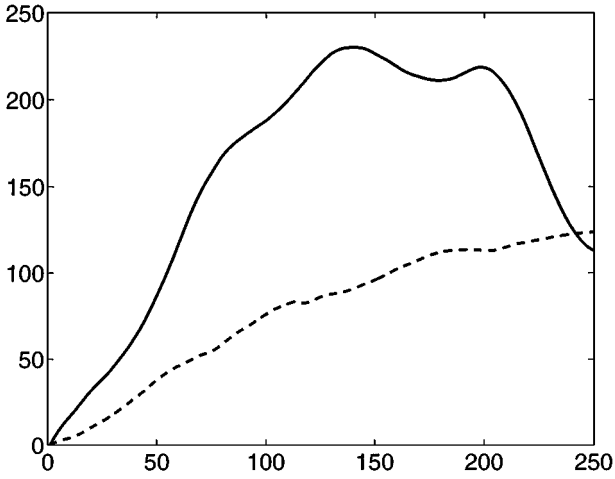

FIG. 4. Displacement of the center of the area of interest in the "rocks" sequence with a static camera (number of iterations in abscissa, pixels in ordinate).

Finally, the controlled rotational speeds, presented in Fig. 7, account for the displacement of the area of interest in the image, meaning, mainly, a horizontal motion with a change of direction in the course of the sequence. This is translated by a rotation around the y axis of the camera, with a changing sign, and a weak speed around the x axis.

On each image of the sequence acquired during the stabilization and presented in Fig. 3, we can notice that the area of interest, meaning the area defined by the first image, is well tracked and remains centered. Very little difference can be seen from one image to another due to translations of the ROV forward and backward. Indeed, the effects of such motions, providing, in the best case when the background is planar, an affine transformation of the image, cannot be compensated for by a pan and tilt control. Nevertheless, the fixation on the same point is maintained from the beginning to the end of the sequence. An important thing to notice is the poor quality of the images and especially the effects of shooting a TV

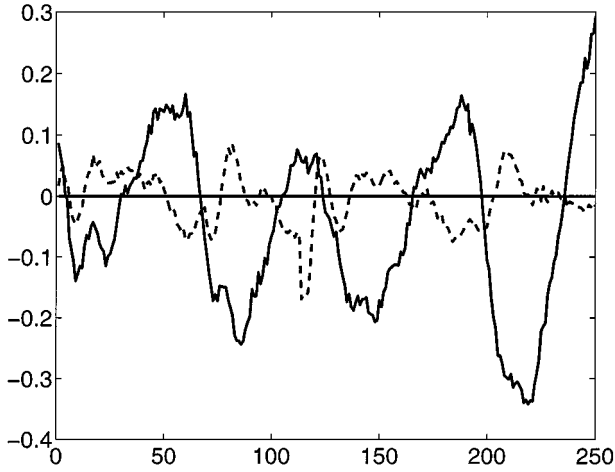

FIG. 5. Displacement of the center of the area of interest center in the "rocks" sequence during stabilization (number of iterations in abscissa, pixels in ordinate).

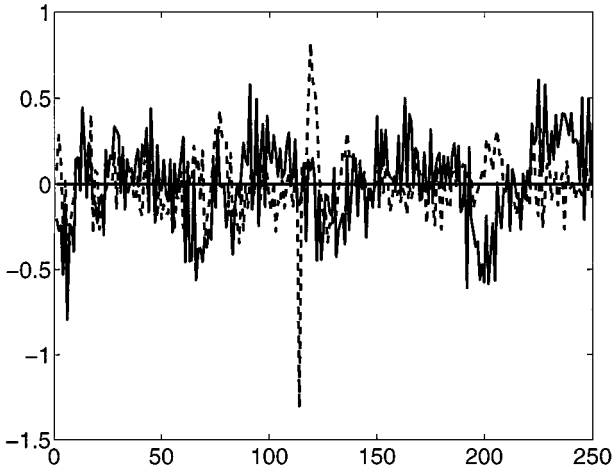

FIG. 6. Speed of motion of the center of the area of interest in the "rocks" sequence during stabilization (number of iterations in abscissa, pixels/s in ordinate).

screen with the CCD camera. Many lines appear, mainly in the horizontal direction, due to the sweeping. However, the estimation of the dominant motion is not perturbed. Another point to notice to emphasize the accuracy of the RMR algorithm is the weak texture of the area of interest. Finally, the illumination is also changing along the task, the image getting darker. Once again, the RMR algorithm is robust with respect to this kind of problem.

5.2. "Sharks" Sequence Results

In the "sharks" sequence, displayed in Fig. 8, the area of interest, the head of one of the sharks, also undergoes a large motion in the image. Departing from the center, it goes from the upper left corner to end in the bottom left corner. (See Fig. 9 for a sequence allowing visual validation of numerical results obtained.) Once again, this is confirmed by

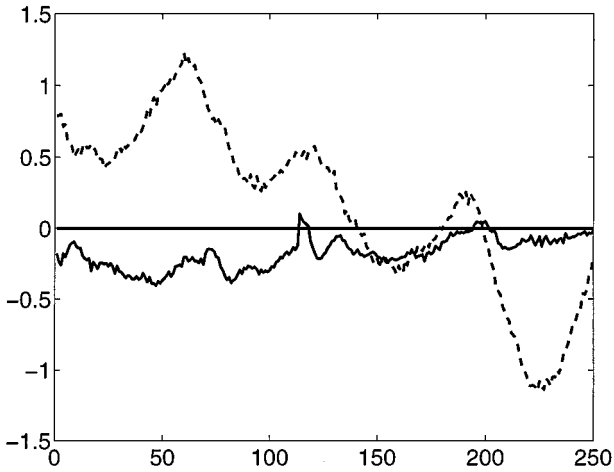

FIG. 7. Computed controlled rotation speeds for the "rocks" sequence during stabilization (number of iterations in abscissa, °/s in ordinate).

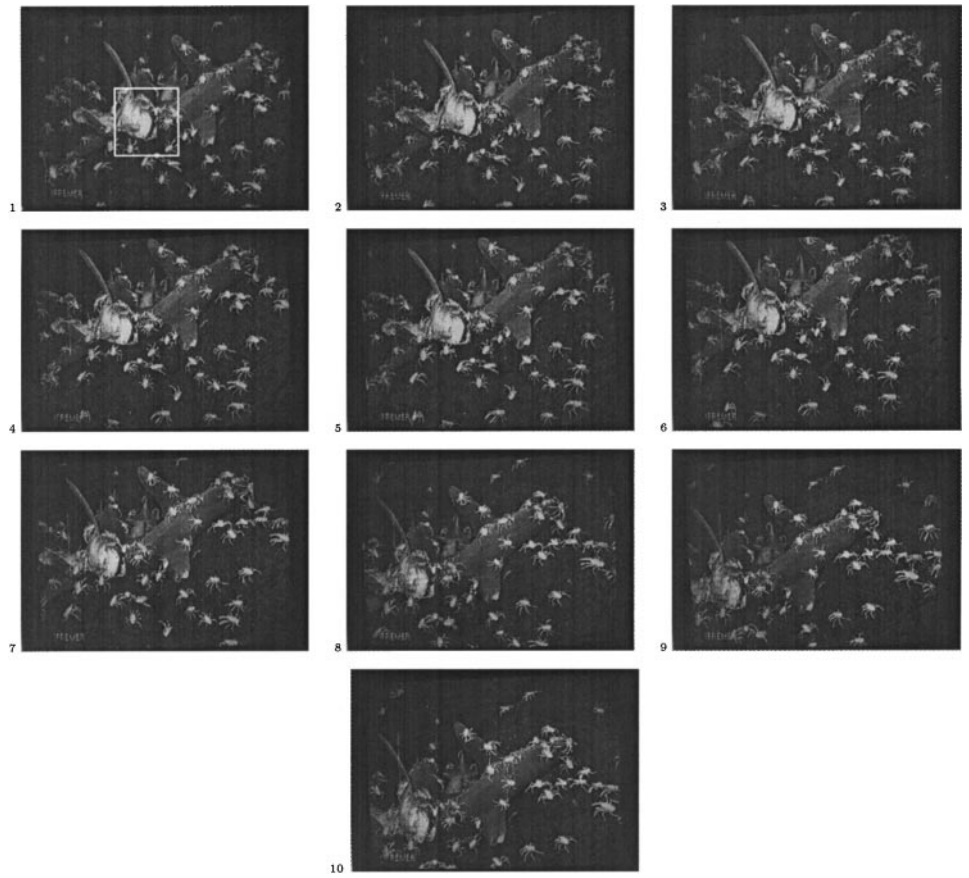

FIG. 8. "Sharks" sequence. One full image out of 25 from the original sequence (one image every 2 s) with a motionless camera.

the displacement of the area of interest when no control is performed to rotate the camera, as is shown in Fig. 10. For this sequence, the amplitude of the motion is more than 100 pixels in the x direction and about 70 pixels in the y direction. The speed goes up to 15 pixels/s along each axis. We can also notice a strong variation of the global illumination along the sequence.

As in the "rocks" sequence, the curves related to the estimated position of the point of interest, displayed in Fig. 11, show that the maximal displacement of this point from the center is less than half a pixel. This value remains thus very weak compared to the displacement which occurs with a static camera. The measured global velocity in the image, presented in Fig. 12, also remains weak, less than 1.5 pixel/s, and even less than 0.5 pixel/s when the velocity is the highest with a static camera (between 10 and 15 pixels/s after iteration 100). Finally, the computed rotational speeds of the camera, sent to the pan and tilt actuators, are in acquaintance with the motion of the area of interest in the image (see Fig. 13). This means a large oscillation around the initial position and then a translation toward the bottom left of the image.

The sequence displayed in Fig. 9 allows us to visually validate these numerical results. The only visible deformation of the image, appearing in the last images, is a small contraction, due to a backward translational motion of the ROV along the optical axis. This

FIG. 9. One image out of 10 of the "sharks" sequence acquired during stabilization (one image every 800 ms).

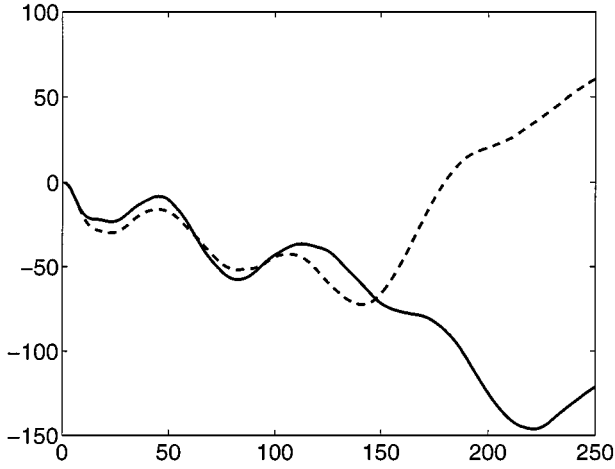

FIG. 10. Displacement of the center of the area of interest in the "sharks" sequence with a static camera (number of iterations in abscissa, pixels in ordinate).

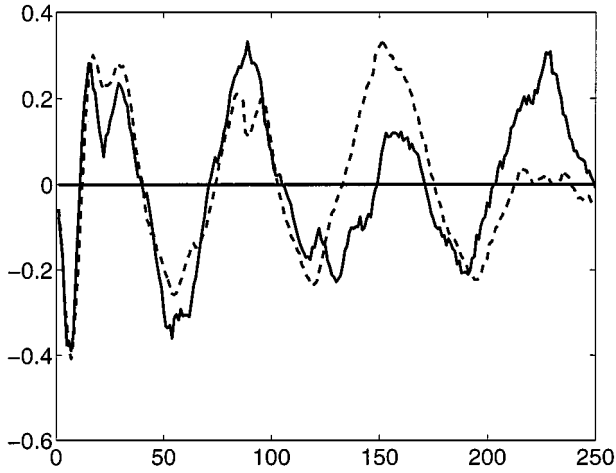

FIG. 11. Displacement of the center of the area of interest in the “sharks” sequence during stabilization (number of iterations in abscissa, pixels in ordinate).

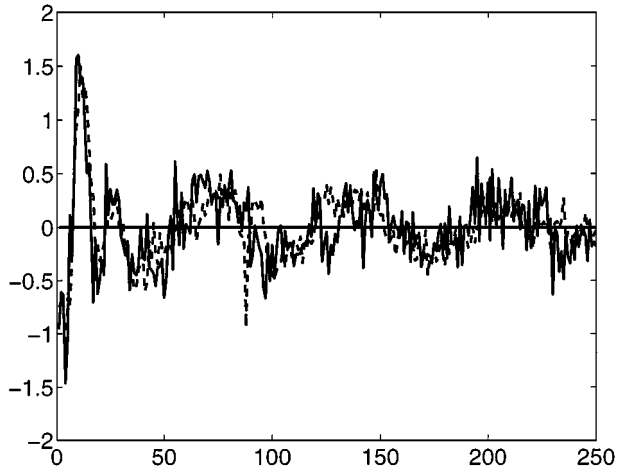

FIG. 12. Speed of the center of the area of interest in the “sharks” sequence during stabilization (number of iterations in abscissa, pixels/s in ordinate).

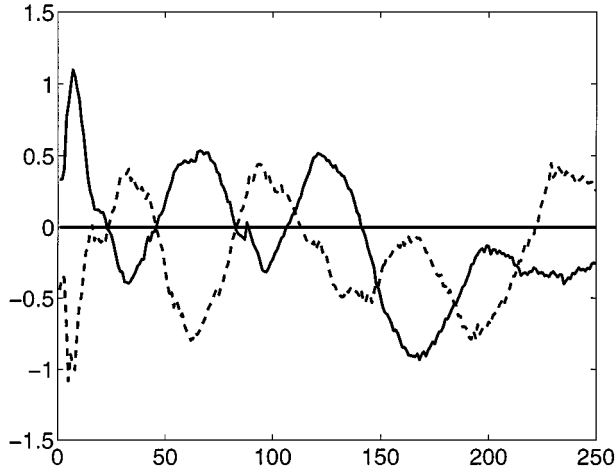

FIG. 13. Computed controlled rotation speeds for the “sharks” sequence during stabilization (number of iterations in abscissa ◦/s in ordinate).

generates a zoom effect which cannot be compensated for by our scheme. Nevertheless, the initial central point is always maintained in the same position. This is done despite two different perturbations. The first is the change of illuminating conditions during the experiment, from lighter to darker. The other is the presence of local motions in the area of interest, here due to the small white crabs. As the RMR algorithm has a step of rejection of local motions that are not coherent with the global motion, this does not perturb the estimation.

6. CONCLUSION

The objective of this study was to validate a stabilization scheme for the acquisition of submarine images with a camera mounted inside a ROV. More precisely, as only the two rotational d.o.f. of the camera corresponding to pan and tilt were used, the aim was to maintain the same point in the middle of the image all along the task. As no a priori information about the visual aspect of the observed scene was provided, it was impossible to perform such an application using pattern-recognition-based image processing. Our idea was to measure the global motion in the image and to integrate it in order to retrieve the position of the point initially in the center of the image.

Before integrating our whole scheme in a ROV, we had to validate it. This was done using a pan and tilt camera observing a TV screen on which previously acquired submarine images were projected. Results presented in this paper show that our approach is accurate both from the control and from the motion estimation perspective. In particular, it shows that no drift appears during stabilization, even if some difficulties arise for image processing. First, there are some technical problems due to the observation of a TV screen. The sweep on the screen produces unexpected lines in the image. Moreover, the estimated motions are complex, as they are due to the reaction of the ROV to water currents, and no knowledge of their direction and amplitude is available. A usual problem for motion estimation is also the need for a sufficient texture. As we rely only on the constant parameters of 2D motion, accurate results are obtained even with weakly textured images. Finally, some local motions can be seen in the area of interest, for example, in the "sharks" sequence. Once again, the RMR algorithm is robust with respect to that kind of perturbation.

Nevertheless, some points are still to be developed. The RMR algorithm has already shown its accuracy for the estimation of more complex models than the constant one (see for example [14] where the quadratic parameters are used within a visual servoing scheme). Thus, we can think of using a more complex model in order to perform a more accurate stabilization. This means compensating not only for the dominant motion in the image, but also for the affine and quadratic deformations by maintaining the relative position between the observer and the scene constant, controlling the 6 d.o.f. of the camera. One of the limitations is now the computational time to obtain such a model with sufficient precision, which is around 1 s and is too large for the system to react smoothly.

ACKNOWLEDGMENTS

This work has received the financial support of Ifremer, the French Research Institute for the Exploitation of the Sea. This article was prepared while Armel Crétual was a postdoctoral fellow in LIRA–Lab at Genoa University, Italy. The authors thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

1. F. Bensalah and F. Chaumette, Compensation of abrupt motion changes in target tracking by visual servoing, in *IEEE Int. Conf. on Intelligent Robots and Systems, Pittsburgh, August 1995*, Vol. 1 pp. 181–187.
2. P. I. Corke and M. C. Good, Controller design for high performance visual servoing, in *12th World Congress IFAC, Sydney, Australia, July 1993*, Vol. 9, pp. 395–398.
3. K. Hashimoto, T. Ebine, K. Sakamoto, and H. Kimura, Full 3D visual tracking with nonlinear model-based control, in *American Control Conference, San Francisco, California, June 1993*, pp. 3180–3185.
4. N. P. Papanikolopoulos, B. Nelson, and P. K. Khosla, Six degree-of-freedom hand/eye visual tracking with uncertain parameters, *IEEE Trans. Robo. Automat.* **11**(5), 1995, 725–732.
5. M. G. P. Bartholomeus, B. J. A. Kröse, and A. J. Noest, A robust multi-resolution vision system for target tracking with a moving camera, in *Computer Science in the Netherlands* (H. Wijshof, Ed.), pp. 52–63, CWI, Amsterdam, 1993.
6. P. Nordlund and T. Uhlin, Closing the loop: Detection and pursuit of a moving object by a moving observer, *Image Vision Comput.* **14**(4), 1996, 265–275.
7. D. Murray and A. Basu, Motion tracking with an active camera, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(5), 1994, 449–459.
8. I. D. Reid and D. W. Murray, Active tracking of foveated feature clusters using affine structure, *Internat. J. Comput. Vision* **18**(1), 1996, 41–60.
9. P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, Automated tracking and grasping of a moving object with a robotic hand-eye system, *IEEE Trans. Robot. Automat.* **9**(2), 1993, 152–165.
10. C. Brown, Gaze control cooperating through prediction, *Image Vision Comput.* **8**(1), 1990, 10–17.
11. E. Milios, M. Jenkin, and J. Tsotsos, Design and performance of TRISH, a binocular robot head with torsional eye movements, *Internat. J. Pattern Recog. Artificial Intell.* **7**(1), 1993, 51–68.
12. D. W. Murray, K. J. Bradshaw, P. F. Mc Lauchlan, I. D. Reid, and P. M. Sharkey, Driving saccade to pursuit using image motion, *Internat. J. Comput. Vision* **16**(3), 1995, 205–228.
13. F. Panerai and G. Sandini, Oculo-motor stabilization reflexes: Integration of inertial and visual information, *Neural Networks* **11**(7/8), 1998, 1191–1204.
14. A. Crétual and F. Chaumette, Positioning a camera parallel to a plane using dynamic visual servoing, in *IEEE Int. Conf. on Intelligent Robots and Systems, Grenoble, France, September 1997*, Vol. 1, pp. 43–48.
15. P. Questa, E. Grossmann, and G. Sandini, Camera self orientation and docking maneuver using normal flow, in *SPIE AeroSense'95, Orlando, Florida, April 1995*.
16. V. Sundareswaran, P. Bouthemy, and F. Chaumette, Exploiting image motion for active vision in a visual servoing framework, *Internat. J. Robot. Research* **15**(6), 1996, 629–645.
17. M. Irani, B. Rouso, and S. Peleg, Recovery of ego-motion using image stabilization, in *Int. Conf. on Computer Vision and Pattern Recognition, Seattle, WA, June 1994*, pp. 454–460.
18. C. Morimoto and R. Chellapa, Fast electronic digital image stabilization, in *IEEE Int. Conf. on Robotics and Automation, Minneapolis, MN, April 1996*, pp. 284–288.
19. F. Spindler and P. Bouthemy, Real-time estimation of dominant motion in underwater video images for dynamic positioning, in *IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, May 1998*, Vol. 2, pp. 1063–1068.
20. L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum, and D. Mishra, Image alignment for precise camera fixation and aim, in *Int. Conf. on Computer Vision and Pattern Recognition, Santa Barbara, June 1998*, pp. 594–600.
21. A. Crétual and F. Chaumette, Image-based visual servoing by integration of dynamic measurements, in *IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, May 1998*, Vol. 3, pp. 1994–2001.
22. S. Negahdaripour, X. Xu, and L. Jin, Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms, *IEEE J. Oceanic Engg.* **24**(3), 1999, 370–382.
23. M. Subarrao and A. Waxman, Closed-form solutions to image equations for planar surface in motion, *Comput. Vision Graphics Image Process.* **36**, 1986, 208–228.

24. J. M. Odobez and P. Bouthemy, Robust multiresolution estimation of parametric motion models, *J. Visual Commun. Image Represent.* **6**, 1995, 348–365.
25. B. Espiau, F. Chaumette, and P. Rives, A new approach to visual servoing in robotics, *IEEE Trans. Robot. Automat.* **8**(3), 1992, 313–326.
26. A. Crétual, F. Chaumette, and P. Bouthemy, Complex object tracking by visual servoing on 2D image motion, in *IAPR Int. Conf. on Pattern Recognition, Brisbane, Australia, August 1998*, Vol. 2, pp. 1251–1254.
27. F. Chaumette and A. Santos, Tracking a moving object by visual servoing, in *12th World Congress IFAC, Sydney, Australia, July 1993*, Vol. 9, pp. 409–414.
28. A. H. Hunt and A. C. Sanderson, *Vision-Based Predictive Robotic Tracking of a Moving Target*, Technical Report CMU-RI-TR-82-15, Department of Electrical Engineering and The Robotics Institute, Carnegie Mellon University, Pittsburgh, 1982.