Armel Crétual
François Chaumette

IRISA/INRIA Rennes
Campus de Beaulieu
35042 Rennes cedex, France
armel.cretual@irisa.fr

Application of Motion-Based Visual Servoing to Target Tracking

Abstract

In this paper, the classical task of mobile target tracking using a pan-and-tilt camera is considered. The authors use recent results in motion-based visual servoing to deal with complex targets for which shape and texture are unknown. The first method consists of designing a control law directly from the estimated image motion. This leads to the computation of the pan-and-tilt acceleration necessary to reduce the tracking error. A second method, more efficient for target tracking, consists of retrieving the target position in the image from its estimated motion. This leads to classical image-based visual servoing. For both methods, experimental results obtained at video rate are presented and discussed.

KEY WORDS—visual servoing, 2D image motion, target tracking, image stabilization

1. Introduction

Target tracking using a pan-and-tilt camera is one of the oldest tasks that has been studied in the robotics and vision communities. The main applications were in the military domain (Gilbert et al. 1980; Dzialo and Schalkoff 1986) or in the design of active stereoscopic heads (Bajcsy 1988; Clark and Ferrier 1989). Researchers dealing with the robotics aspect were generally not interested in vision issues but in control strategy. As a consequence, there was often a strong a priori knowledge of the observed object to validate the control law. Most of these works (Bensalah and Chaumette 1995; Corke and Good 1993; Hashimoto et al. 1993; Papanikolopoulos, Nelson, and Khosla 1995) use a quasi-binary image to easily separate the target from the background. In Reid and Murray (1996), a corner detection algorithm yields the position of a particular point of the target. However, this method is not robust with respect to occlusions. A solution to avoid these problems is to measure the motion in the image. Indeed, such a 2-D motion is independent of the scene content, and targets with complex shape and texture can thus be considered. Now, several algorithms, such as the one presented in Odobez and Bouthemy (1995), are able to perform the estimation of a model of motion in real time, meaning that it is fast enough to be implemented in a robotic loop. This idea has been used, for example, by Yamane, Shirai, and Miura (1998) for person tracking. In fact, the task can be reduced to the detection of the mobile target in front of the background and to the computation of the pan-and-tilt motion to maintain the target in the image center. For the target detection, methods presented in Bartholomeus, Kröse, and Noest (1993) and Nordlund and Uhlin (1996) allow the tracking of a small object or, at best, an object that covers a much smaller part of the image than the background. A 2-D affine motion model is computed between two successive images, and the second image is compensated with the opposite motion. Thresholding the difference between the reconstructed image and the original one gives the position of the target. The method proposed in Murray and Basu (1994) and Murray et al. (1995) is similar, even if compensation is based on the measured motion of the camera, which is obtained using the odometry of the pan tilt head. Larger objects can be tracked, but this method is sensitive to the calibration of the system to make the link between the 3-D motion of the camera and the 2-D projected motion. Finally, in Allen et al. (1993); Brown (1990); Milios, Jenkin, and Tsotsos (1993); and Murray et al. (1995), a stereovision system is used to build a 3-D model of the target motion. However, this approach implies once again having good calibration of the system.

In this paper, we apply to target tracking the two approaches presented in Crétual and Chaumette (2001), which both use an estimation of the target motion in the image. The first approach consists of designing a control law such that the measured motion field reaches a desired one. In that case, tracking a moving target is equivalent to canceling its apparent speed in the image. We will see that this method requires the design of an acceleration-based control law. Furthermore, we

The International Journal of Robotics Research
Vol. 20, No. 11, November 2001, pp. 878-890,

will see that it is not able to completely remove the tracking error. The second approach consists of recovering the target position in the image by successively integrating its estimated speed. In that case, classical image-based visual servoing can be used, and the target tracking task can be translated as observing the coordinates of the target center of gravity at the image center. As a result of the motion estimation algorithm we used, we will see that this method is particularly efficient. It is also robust with respect to calibration errors.

In Section 2, we briefly describe the simplified model of image motion we used. It is adequate for the considered task and allows one to obtain results at video rate. We also describe how the initial target position is simply obtained. The two approaches mentioned above are then presented and discussed in Section 3 and 4, respectively. Experimental results obtained on the pan-and-tilt camera depicted in Figure 1 are included in these sections.

2. Image Processing for Target Tracking

2.1. Image Motion Model

The most classical image motion model used in computer vision has a particular quadratic form with respect to the x and y coordinates of a pixel (see Subbarao and Waxman 1986; Crétual and Chaumette 2001):

$$\begin{cases} \dot{x} &= c_1 + a_1 x + a_2 y + q_1 x^2 + q_2 xy \\ \dot{y} &= c_2 + a_3 x + a_4 y + q_1 xy + q_2 y^2 \end{cases} . \qquad (1)$$

Indeed, this model perfectly represents the image motion field when the camera observes a planar object subject to rigid 3-D motions. More precisely, we have

$$\begin{cases} c_1 = -\frac{T_x}{Z_p} - \Omega_y & c_2 = -\frac{T_y}{Z_p} + \Omega_x \\ a_1 = \gamma_1 \frac{T_x}{Z_p} + \frac{T_z}{Z_p} & a_2 = \gamma_2 \frac{T_x}{Z_p} + \Omega_z \\ a_3 = \gamma_1 \frac{T_y}{Z_p} - \Omega_z & a_4 = \gamma_2 \frac{T_y}{Z_p} + \frac{T_z}{Z_p} \\ q_1 = -\gamma_1 \frac{T_z}{Z_p} - \Omega_y & q_2 = -\gamma_2 \frac{T_z}{Z_p} + \Omega_x \end{cases} , \qquad (2)$$

where T_x, T_y, T_z, Ω_x, Ω_y, and Ω_z respectively represent the three translational and the three rotational components of the kinematic screw between the camera frame and the object frame, with $Z = Z_p + \gamma_1 X + \gamma_2 Y$ being the equation of the object plane expressed in the camera frame. Of course, other motion models may be used. The most usual ones are the constant model (the restriction to terms c_i) and the affine model (the restriction to terms c_i and a_i).

Whatever the chosen model may be, its parameters are computed using the robust multiresolution method presented in Odobez and Bouthemy (1995) and briefly described in Crétual and Chaumette (2001). We just note that it is robust with respect to outliers. Furthermore, it is possible to restrict the estimation on a particular part of the image. In our case, as explained below, we will only consider the projection of the target in the image.

In fact, there is a necessary compromise to find between the accuracy provided by a model and the computation load, such that the control rate is the closest possible to the video rate. Indeed, the real motion in the image is generally complex, and only an approximation can be obtained using a polynomial model. Currently, only the parameters of the constant model can be estimated at the video rate without any dedicated image-processing board. Since the stability of a target-tracking task is directly related to the control rate, we chose to consider only this model.

From the form of c_1 and c_2 given in (2), we note that these parameters contain all the information needed for the tracking task: translational motion T_z along the optical axis and rotational motion Ω_z around the optical axis do not have any influence on c_1 and c_2, but these motions cannot be compensated by a pan-and-tilt camera. All other target motions (T_x, T_y, Ω_x, and Ω_y) will change the value of c_1 or c_2. The control law will thus be able to react to these motions.

Moreover, since we are able to estimate the position of the target in the image (see below), the speed of its center of gravity is given by the approximation at the 0th order of the image motion, that is, (c_1, c_2). No matter which polynomial model of motion is considered, the parameters necessary to realize the tracking are thus only c_1 and c_2.

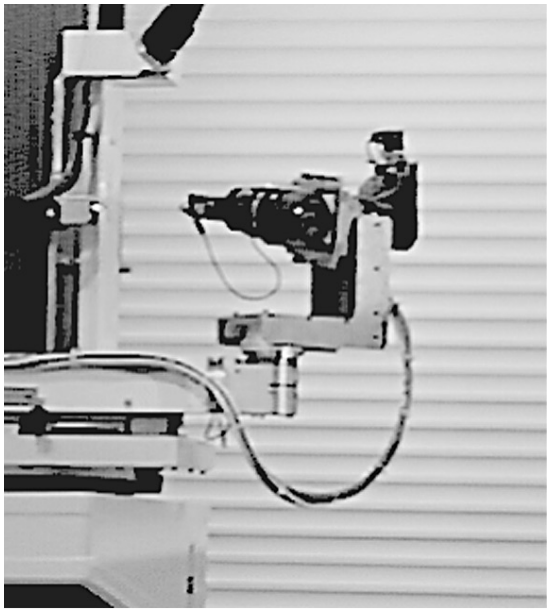

Fig. 1. Experimental cell.

2.2. Initial Detection of the Target

The detection of the target first has to be performed to initialize the tracking. Since we do not exploit any a priori information on the target, this detection step is achieved using the property

that the target undergoes motion. With the camera remaining static until a mobile object is detected, the object location is simply determined by the intensity difference between two successive images. In practice, because of noise in the images, we use a local spatial average of image intensities. Then, by considering a threshold difference between two successive averaged images, we get a binary image separating moving zones from static ones (see Fig. 2). The coordinates (x_0, y_0) at the center of gravity of the detected mask are then easily computed from the binary image.

At the first iteration of the tracking, the estimation of motion parameters c_1 and c_2 is performed in a rectangular window, including the mask, and centered at (x_0, y_0). In this window, points outside the detected mask are considered outliers. Then, at each iteration, the mask and the window are translated with estimated values c_1 and c_2, so that they correspond to the current position of the target.

3. Target Tracking Using 2-D Motion Visual Servoing

We first only consider the estimated terms c_1 and c_2 in the control law. The set of visual features is thus $s = (c_1, c_2)^T$, with desired value $s^* = (0, 0)^T$ so that the target remains static in the image.

3.1. Control Law

From the interaction relation between image motion parameters and camera 3-D motions presented in Crétual and Chaumette (2001), we obtain

$$\begin{aligned} \dot{s} = \begin{pmatrix} \dot{c_1} \\ \dot{c_2} \end{pmatrix} &= L \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t} \\ &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t}, \end{aligned} \tag{3}$$

where $\dot{\Omega}_{c,x}$ and $\dot{\Omega}_{c,y}$ are the camera pan-and-tilt acceleration, L is the interaction matrix related to s, and $\dfrac{\partial s}{\partial t}$ represents the variations of s due to the target's own motion. This equation can be easily obtained from (2).

We thus have the case where the visual features are linked to the controlled camera degrees of freedom through a full-rank invertible matrix. Specifying an exponential decay of s with gain λ ($\dot{s} = -\lambda s$) leads to the following control law:

$$\begin{aligned} \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} &= -L^{-1} \left(\lambda\, s + \widehat{\frac{\partial s}{\partial t}} \right) \\ &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \left(\lambda\, s + \widehat{\frac{\partial s}{\partial t}} \right), \end{aligned} \tag{4}$$

where $\widehat{\dfrac{\partial s}{\partial t}}$ can be obtained from two successive values of s and a measure of the camera pan-and-tilt acceleration (see Crétual and Chaumette 2001 for more details):

$$\widehat{\frac{\partial s}{\partial t}}(t) = \frac{s(t) - s(t - \delta t)}{\delta t} + L \begin{pmatrix} \hat{\Omega}_{c,x}(t) \\ \hat{\Omega}_{c,y}(t) \end{pmatrix}. \tag{5}$$

Furthermore, a classical Kalman filter is used to smooth this estimation.

For the reader interested in the link between the general case presented in Crétual and Chaumette (2001) and the particular case of target tracking described in this paper, we can note that the error vector e is equal to s, meaning combination matrix C (such that $e = C\ (s - s^*)$) is equal to $\mathbb{I}_2$. Furthermore, the interaction matrix L is constant and does not depend on unknown parameters ($\widehat{L} = L$). Finally, from the stability condition exhibited in Crétual and Chaumette (2001) ($K = (CL)(C\widehat{L})^{-1} > 0$), the exponential decrease of $\|s\|$ is ensured at each iteration of the control law, since K is nothing but $\mathbb{I}_2$.

3.2. Results

Images of size 256×256, acquired by a SunVideo board, are processed on an UltraSparc station with a 250 MHz clock. The complete processing rate is about 50 ms per iteration (20 Hz). For the camera used for the experiments, a standard calibration procedure provided the coordinates of the principal point (132, 130.5) and the size of a pixel ($19.8 \times 20.6\ \mu$m).

The experiment was carried out with a textured object from which no geometric features could be easily computed (see Fig. 2). The camera was about 1 m away from the object, which appeared near the image center before it started moving. The object was translating along a rail alternatively to the right and to the left at a constant speed (25 cm/s), with a 4-second pause between the two motion phases. Accelerations and decelerations were performed with an absolute value of 40 cm/s^2, meaning the constant-level speed was reached in 15 iterations. To separate the object from the background, a motion detection step was first performed, as described in Section 2.2.

Two curves related to this experiment are presented in Figure 3. First, the two visual features c_1 and c_2 are given in Figure 3a. Then, the computed control law $\dot{\Omega}_{c,x}$ and $\dot{\Omega}_{c,y}$ sent to the low-level pan tilt controller is given in Figure 3b. Despite some oscillations, the system brings the error to zero after each abrupt change in the object motion. The oscillations are not induced by noisy measurements of the visual features but are a result of a noisy estimation of the object's own motion $\widehat{\frac{\partial s}{\partial t}}$. Indeed, they appear mainly during the acceleration steps, which are very short. Since the noise variance on the state model of the Kalman filter cannot be too low to allow the system to react, it is difficult to obtain more accurate estimations during the abrupt changes of target motions.

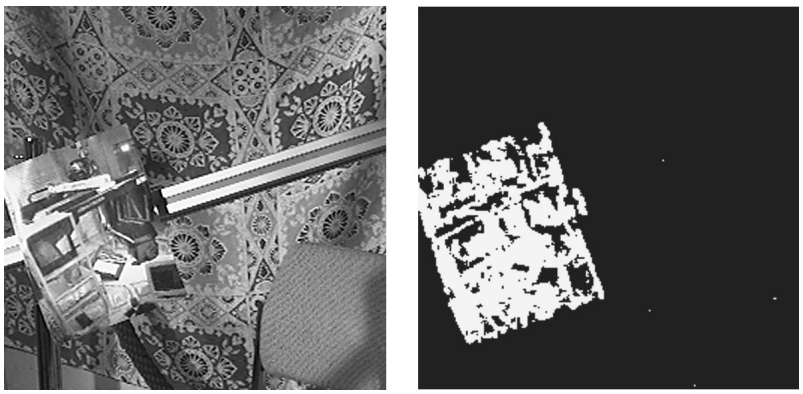

Fig. 2. Detection of the initial target position.

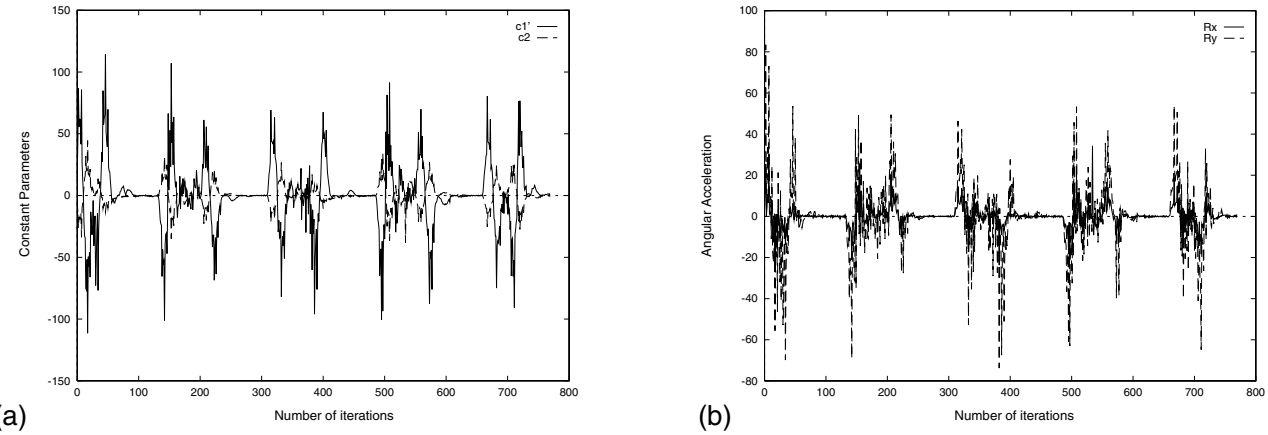

Fig. 3. Target tracking using the motion parameters directly: (a) constant parameters c_1 and c_2 and (b) angular acceleration (in deg/s^2).

Nevertheless, to show the accuracy of the method, we present in Figure 4 the two coordinates at the center of gravity of the target in the image. These values are computed by a simple integration of the speed, using as initialization the center of gravity of the detection mask. They are obviously unused in the presented control scheme.

We can note that a small deviation appears in the object position (approximately 10 pixels after 800 iterations). This deviation is due to the time needed, even if it is short, to bring the constant parameters to zero. Using a control law directly based on motion parameters, it is of course impossible to completely remove this lag. Furthermore, it is also impossible to reach a desired position of the target in the image. That is why we propose, in the next section, using a control law based on the estimated position of the target in the image.

To conclude this part, let us emphasize that even if the use of motion parameters directly in the control loop does not seem to be the most efficient method for target tracking, this approach is sometimes the only one possible for more complex robotics tasks, as demonstrated in Crétual and Chaumette (2001).

4. Target Tracking by Integration of 2-D Motion

We now consider as input of the control law the estimated position of the target position in the image. The visual features are thus $s = (x, y)^T$. They are simply obtained by successive summations of the estimated velocities c_1 and c_2:

$$s_i(k) = s_i(0) + \sum_{i=1}^{k} c_i \ \delta t, \tag{6}$$

where $s_i(0) = (x_0, y_0)^T$ is the initial position of the target computed during the detection step (see Section 2.2), and δt is the period of the control loop.

The aim of the tracking task is to control the pan-and-tilt camera such that the image of the mobile target first is brought to the image center ($s^* = (0, 0)^T$) and then remains at this position no matter what the target motions are. This task is quite simple from the control point of view. Our contribution is more concerned with the complexity of the considered targets.

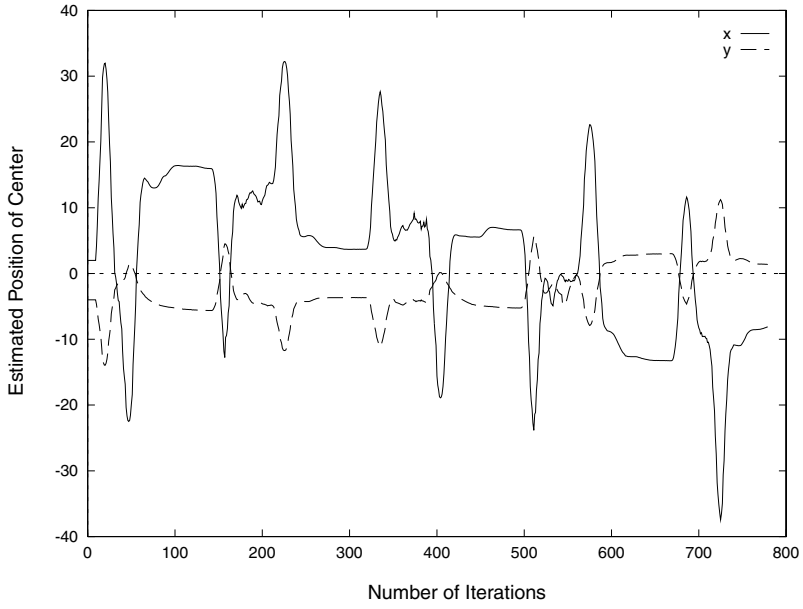

Fig. 4. Object center position.

4.1. Control Law

To design the control law, we use the relation between the temporal variation of s and the controlled camera motion, that is, $\Omega_{c,x}$ and $\Omega_{c,y}$ around the x and y axes. We get the following from the classical equation between image point velocity and 3-D kinematic screw (Espiau, Chaumette, and Rives 1992; Hutchinson, Hager, and Corke 1996; Crétual and Chaumette 2001):

$$\dot{s} = L \begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t} \text{ with } L = \begin{bmatrix} xy & -(1+x^2) \\ (1+y^2) & -xy \end{bmatrix}. \tag{7}$$

We can see that the interaction matrix L only depends on the estimated position of the target in the image. Assuming this estimation is accurate, one can set $\widehat{L} = L$. Therefore, specifying as in the previous case an exponential decrease of s leads to the following (Crétual and Chaumette 2001):

$$\begin{aligned} \begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} &= -\lambda\, L^{-1} s - L^{-1} \widehat{\frac{\partial s}{\partial t}} \\ &= -\frac{\lambda}{1+x^2+y^2} \begin{pmatrix} y \\ -x \end{pmatrix} - L^{-1} \widehat{\frac{\partial s}{\partial t}}. \end{aligned} \tag{8}$$

In practice, the chosen gain λ is not constant but adaptive. More precisely, this gain is set to a low value when the error is great, to preserve the stability of the system. However, it is increased to a high value when the error is weak to optimize the time to convergence.

Furthermore, as explained in Chaumette and Santos (1993), the estimation $\widehat{\frac{\partial s}{\partial t}}$ of $\frac{\partial s}{\partial t}$ can be obtained from (7):

$$\widehat{\frac{\partial s}{\partial t}} = \widehat{\dot{s}} - L\, \widehat{\Omega_c}. \tag{9}$$

In our case, the measure $\widehat{\dot{s}}$ of $\dot{s}$ is directly supplied by the image motion estimation algorithm ($\widehat{\dot{s}} = (c_1\ c_2)^T$), while $\widehat{\Omega_c}$ is the measured camera rotational velocity.

We can note that if the target is motionless, a first-order approximation gives

$$s_{(k)/(k-1)} = s_{k-1} + \dot{s}\ \delta t_k = s_{k-1} + L\, \widehat{\Omega_c}\ \delta t. \tag{10}$$

Thus, the estimation of the target's own motion in the image at iteration k can be written as

$$\left(\widehat{\frac{\partial s}{\partial t}}\right)_{(k)} = \frac{s_{(k)} - s_{(k)/(k-1)}}{\delta t}. \tag{11}$$

This value represents the discrepancy between the actual measure of the visual feature and the predicted one. It is null if the target is motionless and constant if the target velocity is also constant. This means that the target motion and the discrepancy have the same model.

Let us now return to the control point of view and 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 camera velocity measurement errors.

According to several works investigating the field of filtering for target tracking (Blackman 1986; Hunt and Sanderson 1982; Corke and Good 1993; Allen et al. 1993), two

common approaches are employed: the first consists of using fixed tracking coefficients ($\alpha - \beta$, $\alpha - \beta - \gamma$ trackers), and the second, Kalman filtering, generates time-variable tracking coefficients that are determined by a priori models of target dynamics. While the first approach has computational advantages, the second one seems much more appealing because of the adaptability of its coefficients for tracking highly maneuvering targets. However, implementing a Kalman filter requires first defining or estimating the state model evolution of the parameters, with the simplest cases for motion parameters being the constant speed and constant acceleration models.

When a target maneuvers (e.g., when abrupt changes in its acceleration occur), a tracking filter should respond. Such maneuvering may be detected by a rapid increase in the normalized discrepancy. The recommended methods for dealing with those situations are numerous (Bensalah and Chaumette 1995; Brown et al. 1989), and we have chosen, for robustness issues, to consider model maneuvers as "colored noise." That is why we have chosen a constant acceleration state model with colored noise, the equations of which are given by

$$\left\{ \begin{array}{rcl} \left(\dfrac{\partial s}{\partial t}\right)_{(k+1)} & = & \left(\dfrac{\partial s}{\partial t}\right)_{(k)} + \Delta t \left(\dfrac{\dot{\partial s}}{\partial t}\right)_{(k)} + \nu_{(k)} \\ \nu_{(k+1)} & = & \rho\ \nu_{(k)} + \upsilon_{1(k)} \\ \left(\dfrac{\dot{\partial s}}{\partial t}\right)_{(k+1)} & = & \left(\dfrac{\dot{\partial s}}{\partial t}\right)_{(k)} + \upsilon_{2(k)} \end{array} \right. , \tag{12}$$

where ρ is the degree of correlation between successive accelerations and can range from 0 to 1 (0.3 in the experiments described below), and υ_1 and υ_2 are the zero-mean Gaussian white noise values on the chosen model. Furthermore, the relation involved in the Kalman filter relating the observed data to the chosen model is given by

$$\left(\widehat{\frac{\partial s}{\partial t}}\right)_{(k)} = \left(\frac{\partial s}{\partial t}\right)_{(k)} + \omega_{(k)}, \tag{13}$$

where $\widehat{\dfrac{\partial s}{\partial t}}$ is the estimated value given by (9), and ω is a zero-mean Gaussian white noise on the observations.

Finally, let us note that the control law given by (8) 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, is used; this leads to the following adaptive predictive control law:

$$\begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} = -\frac{\lambda}{1 + x^2 + y^2} \begin{pmatrix} y \\ -x \end{pmatrix} - L^{-1} \left(\widehat{\frac{\partial s}{\partial t}}\right)_{(k+1)/(k)}. \tag{14}$$

4.2. Sensitivity to Calibration Errors

In several articles (Murray and Basu 1994; Murray et al. 1995), detecting the object of interest in a tracking scheme is based on the known motion of the camera. Indeed, it is well known that if a rotating camera is observing a fixed scene, the apparent motion field in the image is closed to a constant one. More precisely, the four affine parameters a_i that appear in eq. (1) are equal to zero. Moreover, in such a case, there is no influence of the target depth on its apparent motion. It is thus possible to compensate the camera motion in the image from the measure of the camera rotation. Therefore, under the hypothesis that only one object is moving in the scene, a simple difference between the compensated image and the one acquired at the current iteration gives the position of the object.

However, this approach necessitates a good calibration of the system. First, the center of rotation must be the optical center of the camera. If this is not the case, camera translational motions are also performed, and the depth of each point of the scene appears in the 2-D motion equations. In the best-case scenario, when the induced translation is known, these depths remain unknown. Therefore, some perturbations occur in the compensation, especially if there are some great depth changes in the observed scene. Another point is that the computation of the 2-D motion in the image from 3-D motion necessitates precise knowledge of the intrinsic parameters of the camera (size of pixel, position of the principal point, and radial distortion). When inaccurate intrinsic parameters are used, the image compensation cannot be performed correctly, and it thus becomes impossible to separate the object of interest from the background.

In our approach, the measured camera velocity is only used in the control scheme to reduce the tracking errors (see eq. (9)). If a wrong estimation of the camera velocity occurs due to a coarse camera calibration, it will have an effect on the response of the controller, but nothing more, since the measured camera velocity is not used to determine the target position in the image.

Furthermore, the motion parameters c_1 and c_2, used in the control law to estimate the target position, are metric (expressed in m/s and rad/s). However, the image motion estimation algorithm we use to compute these parameters provides results expressed in pixels/s. Therefore, it is necessary to distinguish the former from the latter. The intrinsic parameters of the camera, or at least a coarse estimation of these parameters, should then be used. Nevertheless, we will show that our control scheme is not sensitive to calibration errors.

Indeed, if we denote with a p index the parameters of the motion model expressed in pixels, we have the corresponding relation with the metric ones (Crétual 1998):

$$\begin{pmatrix} c_1 \\ c_2 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ q_1 \\ q_2 \end{pmatrix} = M_{m2p} \begin{pmatrix} c_{1_p} \\ c_{2_p} \\ a_{1,p} \\ a_{2_p} \\ a_{3,p} \\ a_{4,p} \\ q_{1,p} \\ q_{2,p} \end{pmatrix}$$

with $M_{m2p} =$

$$\begin{bmatrix} l_x & 0 & l_x d_x & l_x d_y & 0 & 0 & l_x d_x^2 & l_x d_x d_y \\ 0 & l_y & 0 & 0 & l_y d_x & l_y d_y & l_y d_x d_y & l_y d_y^2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2d_x & d_y \\ 0 & 0 & 0 & l_x/l_y & 0 & 0 & 0 & d_x l_x/l_y \\ 0 & 0 & 0 & 0 & l_y/l_x & 0 & d_y l_y/l_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & d_x & 2d_y \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/l_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/l_y \end{bmatrix}$$

where (l_x, l_y) is the ratio between the camera focal length and the size of a pixel, and (d_x, d_y) is the distance between the principal point and the center of the estimation window. Since the principal point is generally close to the image center, we have in the neighborhood of convergence $(x, y) \approx (0, 0)$ and $(d_x, d_y) \approx (0, 0)$. We thus obtain

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \approx \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \approx \begin{pmatrix} l_x\ c_{1,p} \\ l_y\ c_{2,p} \end{pmatrix}.$$

We can see that an error on the position of the principal point has no effect. Furthermore, a wrong estimation of (l_x, l_y) only provides a scale factor error. Since the control law has a closed-loop structure, this does not perturb the convergence. In the worst case, the convergence is simply obtained more slowly. It can even be obtained more quickly, since the influence of such an error is more or less like having a scale factor on the decreasing gain λ. We will demonstrate the effective robustness of our method with respect to calibration errors in the next section.

4.3. Results

4.3.1. Tracking a Rigid Object

The same object as in Section 3.2 is first considered. It again translates at a constant speed along a rail alternatively to the right and to the left, with a 4-second pause between the two motion phases. First (until iteration 800), the object speed is 8 cm/s, and then it is 30 cm/s. Once again, the camera is about 1 m away from the object, but it does not appear near the center of the image before it starts translating.

The measured position of the target in the image ($s = (x\ y)^T$) is plotted in Figure 5. This experiment shows that convergence is correctly obtained for an initial gap of about 40 pixels (the error is brought to zero in less than 40 iterations, even if the object motion is initially on the opposite direction of the image center). At each abrupt change in the target motion (stop or start), there is an overrun due to the Kalman filter reaction time (approximately 30 pixels for a 30 cm/s speed of the object), but convergence is still obtained and tracking errors are suppressed in few iterations.

In Figure 6, some images are displayed that were acquired during this experiment (1 image in 10). Estimated target position is designated by a cross (+) and the image center by a diamond ($\diamond$). We notice that as soon as the object is brought to the image center (which is already done in image 2), it remains there even if the object stops and restarts. Therefore, from the qualitative point of view, we can conclude that the motion estimation is accurate.

Other experiments, described in Crétual (1998) and using a simpler object from which it was easier to extract the true position, have shown that the difference between the real and the estimated positions was always less than half of a pixel. In those experiments, the object had four markers. The 2-D motion was computed from the displacement of these markers. Of course, the measured 2-D position of the markers was not used in the control loop but was only used to estimate the object center as the center of gravity of the four markers. Moreover, it was also shown from the known object motion at a given distance from the camera that the image motion estimation algorithm gives more equivalent results than the estimation provided by the computation of the four markers. It was even better for slow motion. We can thus conclude that the motion estimation algorithm is unbiased since no drift appears, even after a large number of iterations of the control law.

Another important point to underline is the poor illumination quality (the object is very dark when it is on the right side of the rail) and the fact that this illumination varies. This independence to illumination conditions is due to the robustness of the image motion estimator. We also notice that it seems very difficult to extract the object from the background using only geometric image processing. Finally, even if other objects are moving behind the target from images 8 to 14 and 17 to 25, the tracking system remains stable. This can be explained by the fact that the motion computation relies on the initial mask, and then the estimation is performed only on pixels belonging to the mask. Moreover, the estimation algorithm is robust with respect to potential outliers.

4.3.2. Sensitivity

Several experiments have been done to prove the reliability of our approach in case of weak calibration of the system. The camera has been displaced by approximately 5 cm in two different directions with respect to the pan-and-tilt cell (along the y axis of the image plane and along the optical axis). This means that pan-and-tilt rotations are not performed around the optical center. This implies translational motions of the camera (nearly 10% of the distance between the camera and the target). Moreover, errors have been introduced in the intrinsic parameters of the camera. The principal point has been considered to be the image center (128, 128), and a 10% error has been added on the size of each pixel, increasing to 21.8 $\times$ 22.7 μm. All other conditions were the same as for the previous experiment. Figure 7 displays the estimated target position versus iteration number. We notice that the behavior is very close to that observed in the previous experiment, with

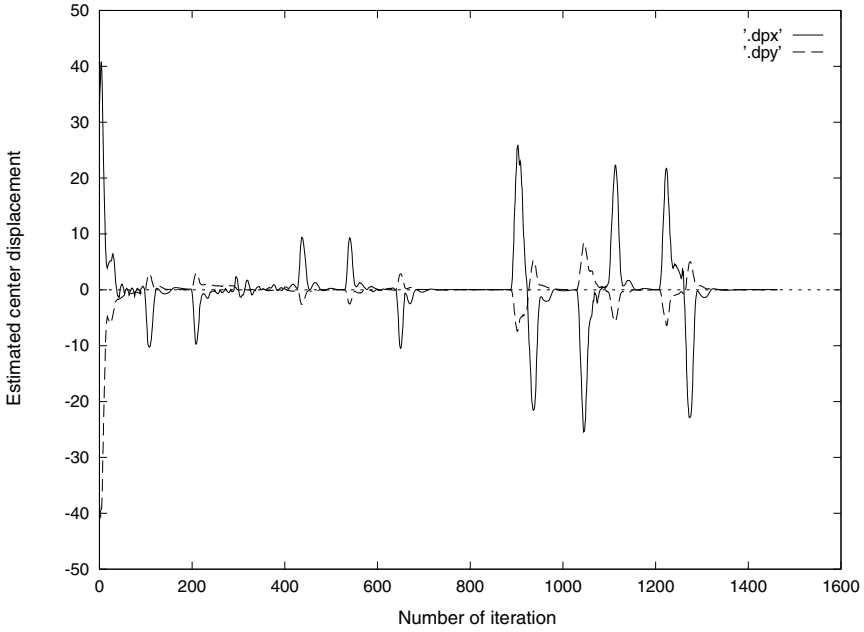

Fig. 5. Estimated target position (in pixel) versus iteration number.

a better calibrated system. The initial error is quickly brought to zero, and the overrun after each change of the object motion is rapidly compensated. These overruns are not higher in this case than previous ones. The only difference is that small oscillations appear (always less than 2 pixels), but they do not generate any divergence. They even decrease in the permanent step.

4.4. Other Applications

The method presented above can be used in other applications. Two of them have been developed with the same control law.

A pedestrian tracking task is presented in Crétual, Chaumette, and Bouthemy (1998). The aim is the same as the application presented above, even if a pedestrian is not a planar rigid object. Let us point out that the estimation of 2-D motion parameters with the algorithm we used involves discarding noncoherent local motions considered as outliers. Therefore, motions related to deformations of nonrigid objects (such as a human being) do not greatly affect the estimation of the dominant motion.

Figure 8a displays the mask obtained from the detection step, in which moving zones appear in white. Figure 8b presents an image acquired during the tracking. The white rectangle represents the including rectangle of the detected mask on which the estimation is performed.

Figure 9 contains 1 image in 10 of the sequence acquired during the tracking. Motion of the person is first sideways and not always facing the camera. Then, the pedestrian comes back to the camera. In each image, the estimated position is represented by a cross (+), and the image center by a diamond ($\diamond$). Despite the complexity of motion, the pedestrian always appears at the image center. This demonstrates the robustness of the motion estimation algorithm and of the control scheme. Small tracking errors appear due to the reaction time of the Kalman filter, but they are always less than 8 pixels. Finally, from images 10 to 13, another person crosses the one being tracked. Despite this perturbing supplementary motion, the camera is still fixating at the selected person.

Finally, in Crétual and Chaumette (2000), we have used the same algorithm for image stabilization of a camera mounted on an underwater engine. In that case, the motion in the image is mostly due to the undesirable motion of the engine, resulting from underwater currents. Moreover, some perturbations can occur in this image motion, due to local motion in the 3D scene, such as fish's motions or, as can be seen in the sequence presented below, due to smoke and gas. Once again, even if the quality of the images used is poor (they had low spatio-temporal gradients), the results presented in Crétual and Chaumette show that the drift in the image remains very weak (less than half a pixel after 250 iterations). A typical image sequence acquired during the stabilization is given in Figure 10, in which the considered scene is a rock from which smoke and gas escape.

5. Conclusion

In this paper, we have presented an application of the two methods presented in Crétual and Chaumette (2001) to perform visual tracking without any a priori knowledge of the

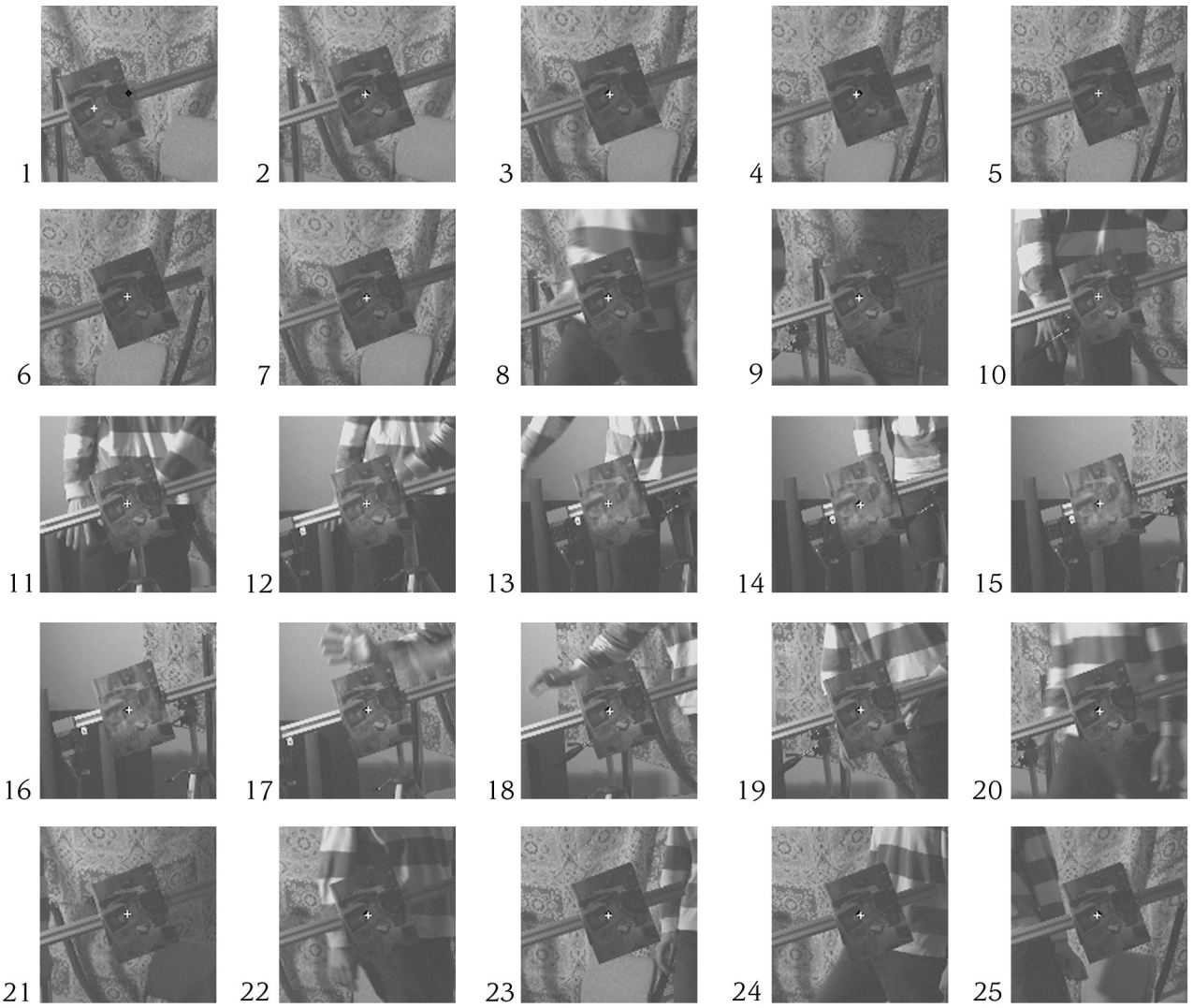

Fig. 6. Tracking a textured object: 1 image in 10 of the acquired sequences (approximately 2 frames per second). + = the estimated target position; ◇ = the image center.

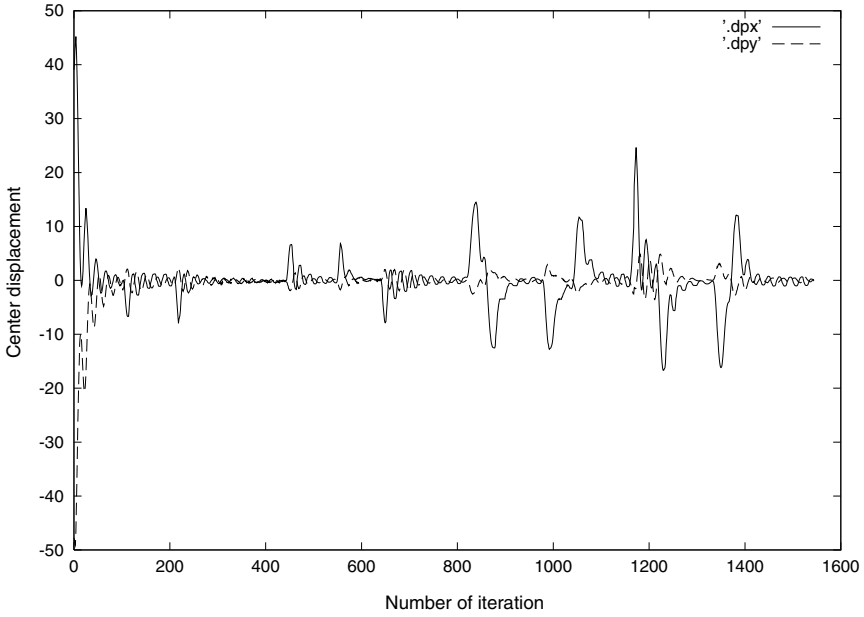

Fig. 7. Estimated target position with a weakly calibrated system.

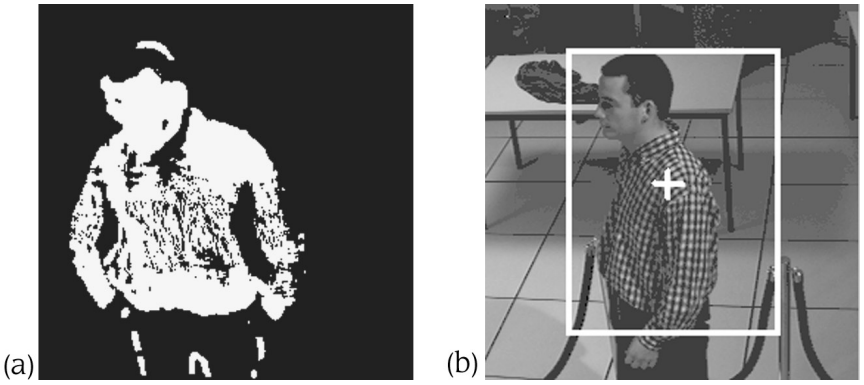

Fig. 8. (a) Detected mask. (b) Estimation window and estimated position of the pedestrian.

object appearance. Both are based on the use of image motion. The principle of the first method is to control the camera motions so that the current image motion reaches a desired value. In the second method, the position of the target is retrieved by successive summation, and classical image-based visual servoing is then used.

These methods were applied to the tracking of real objects, meaning that they were not designed to simplify any visual-processing algorithm. In both cases, satisfying results were obtained, even if the direct use of motion parameters as the input of the control law is less powerful. It thus seems that for visual tasks in which both approaches are applicable, it is more interesting to adopt the second approach (integration of motion). Indeed, the control loop of the second approach can be seen as the combination of the control loop given by the first approach and an integrator. It thus allows us to avoid tracking errors. We can also emphasize that the image motion estimation algorithm we used seems to be unbiased since no drift appears in the experiments.

Improvements could be done on the practical level. In particular, the goal of the tracking experiments presented here was to validate the two control strategies. It explains why the initialization step was based on the strong hypothesis of a single moving target. An image-processing algorithm dedicated to the automatic selection of a target in case of multiple detections should be developed for real survey applications.

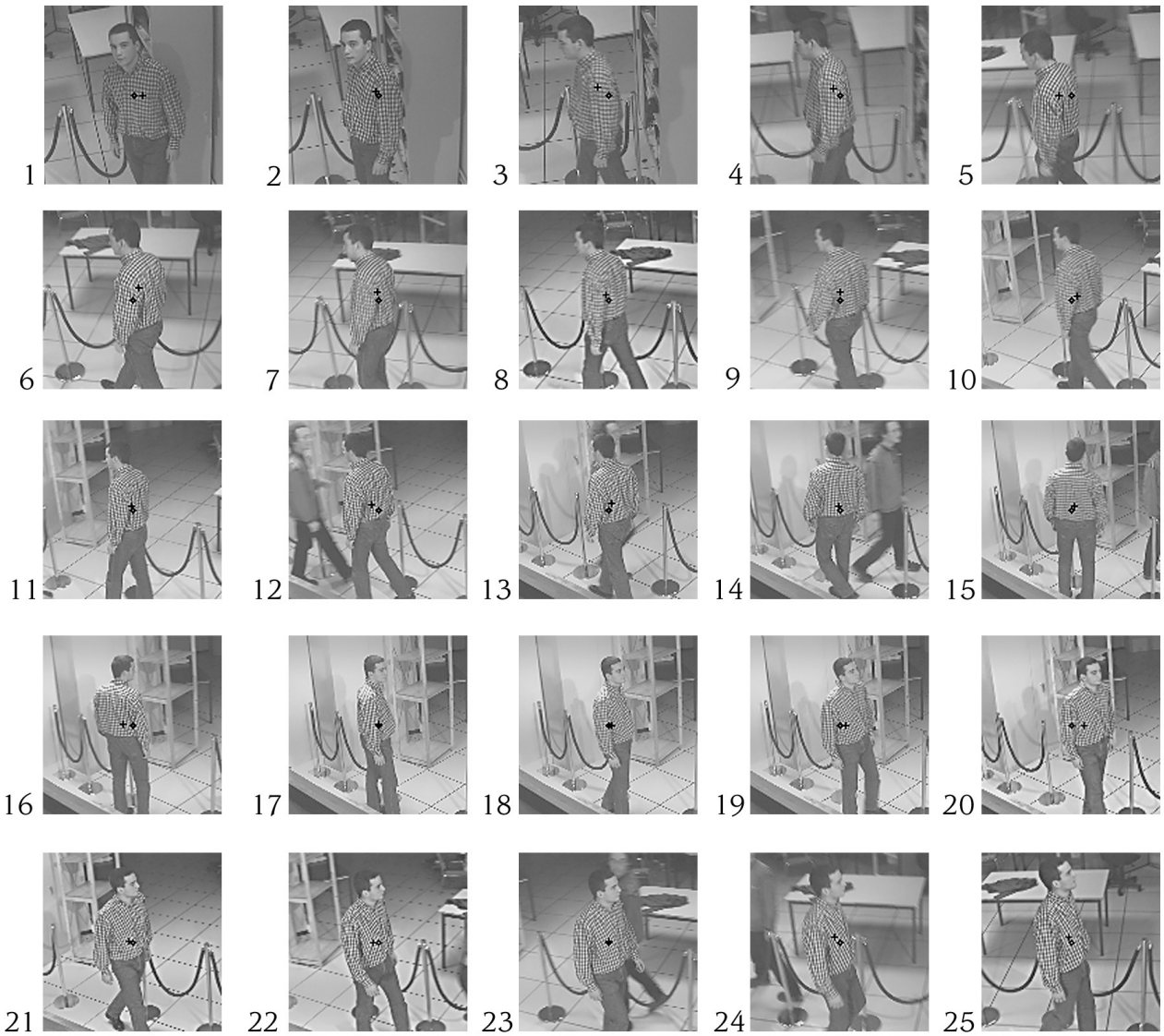

Fig. 9. Tracking of a pedestrian: 1 image in 10 of the acquired sequence (approximately 2 frames per second). + = estimated target position; ◇ = the image center.

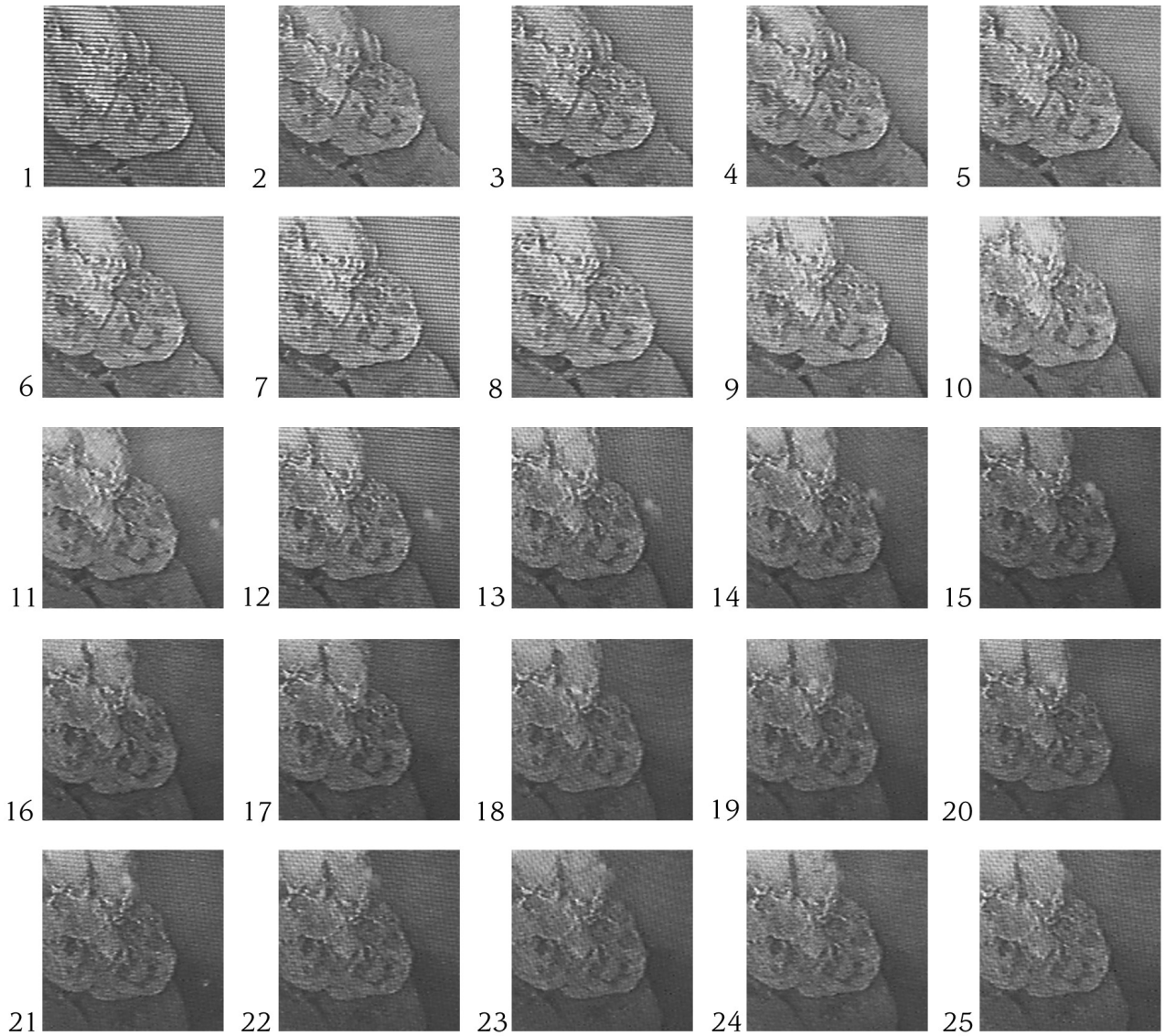

Fig. 10. One image in 10 of a sequence acquired during submarine image stabilization.

References

Allen, P. K., Timcenko, A., Yoshimi, B., and Michelman, P. 1993. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation* 9(2):152–165.

Bajcsy, R. 1988. Active perception. Research Report MS-CIS-88-24, University of Pennsylvania, Philadelphia.

Bartholomeus, M.G.P., Kröse, B.J.A., and Noest, A. J. 1993. A robust multi-resolution vision system for target tracking with a moving camera. In *Computer Science in the Netherlands*, ed. H. Wijshof, 52–63. Amsterdam: CWI.

Bensalah, F., and Chaumette, F. 1995. Compensation of abrupt motion changes in target tracking by visual servoing. *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, Pittsburgh, August, pp. 181–187.

Blackman, S. 1986. *Multiple Target Tracking with Radar Application*. Boston: Artech House.

Brown, C. 1990. Gaze control cooperating through prediction. *Image and Vision Computing* 8(1):10–17.

Brown, C., Durrant-Whyte, H., Leonard, J., and Rao, B. 1989. Centralized and decentralized Kalman filter techniques for tracking, navigation, and control. *Image Understanding Workshop*, SanMateo, CA, May, pp. 651–675.

Chaumette, F., and Santos, A. 1993. Tracking a moving object by visual servoing. *12th World Congress IFAC*, vol. 9, Sydney, Australia, July, pp. 409–414.

Clark, J. J., and Ferrier, N. J. 1989. Control of visual attention in mobile robots. *IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May, pp. 826–831.

Corke, P. I., and Good, M. C. 1993. Controller design for high performance visual servoing. *12th World Congress IFAC*, vol. 9, Sydney, Australia, July, pp. 395–398.

Crétual, A. 1998. *Asservissement visuel à partir d'informations de mouvement dans l'image*. Ph.D. thesis, Rennes 1 University.

Crétual, A., and Chaumette, F. 2000. Dynamic stabilization of a pan and tilt camera for sub-marine image visualization. *Computer Vision and Image Understanding* 79(1):47–65.

Crétual, A., and Chaumette, F. 2001. Visual servoing based on image motion. *International Journal of Robotics Research* 20(11):857–877.

Crétual, A., Chaumette, F., and Bouthemy, P. 1998. Complex object tracking by visual servoing on 2D image motion. *IAPR International Conference on Pattern Recognition*, vol. 2, Brisbane, Australia, August, pp. 1251–1254.

Dzialo, K. A., and Schalkoff, R. J. 1986. Control implications in tracking moving objects using time-varying perspective-projective imagery. *IEEE Transactions on Industrial Electronics* 33(3):247–253.

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

Gilbert, A. L., Giles, K. G., Flachs, G. M., and Rogers, R. B. 1980. A real-time video tracking system. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2(1):47–56.

Hashimoto, K., Ebine, T., Sakamoto, K., and Kimura, H. 1993. Full 3D visual tracking with nonlinear model-based control. *American Control Conference*, San Francisco, June, pp. 3180–3185.

Hunt, A. H., and Sanderson, A. C. 1982. Vision-based predictive robotic tracking of a moving target. Pittsburgh, PA: Department of Electrical Engineering & The Robotics Institute, CMU. Technical Report CMU-RI-TR-82-15.

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

Milios, E., Jenkin, M., and Tsotsos, J. 1993. Design and performance of TRISH, a binocular robot head with torsional eye movements. *International Journal of Pattern Recognition and Artificial Intelligence* 7(1):51–68.

Murray, D., and Basu, A. 1994. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(5):449–459.

Murray, D. W., Bradshaw, K. J., McLauchlan, P. F., Reid, I. D., and Sharkey, P. M. 1995. Driving saccade to pursuit using image motion. *International Journal of Computer Vision* 16(3):205–228.

Nordlund, P., and Uhlin, T. 1996. Closing the loop: Detection and pursuit of a moving object by a moving observer. *Image and Vision Computing* 14(4):265–275.

Odobez, J. M., and Bouthemy, P. 1995. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation* 6(4):348–365.

Papanikolopoulos, N. P., Nelson, B., and Khosla, P. K. 1995. Six degree-of-freedom hand/eye visual tracking with uncertain parameters. *IEEE Transactions on Robotics and Automation* 11(5):725–732.

Reid, I. D., and Murray, D. W. 1996. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision* 18(1):41–60.

Subarrao, M., and Waxman, A. 1986. Closed-form solutions to image equations for planar surface in motion. *Computer Vision, Graphics, and Image Processings* 36(2):208–228.

Yamane, T., Shirai, Y., and Miura, J. 1998. Person tracking by integrating optical flow and uniform brightness regions. *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May, pp. 3267–3272.