IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95
Vol. 1, pp. 181-187, Pittsburgh, USA, August 1995

Compensation of Abrupt Motion Changes in Target Tracking by Visual Servoing

Farabi Bensalah, François Chaumette
IRISA-INRIA Rennes
35042 Rennes cedex, France
e-mail: {bensalah, chaumett}@irisa.fr

Abstract

This paper describes a real time visual target tracking using the generalized likelihood ratio (GLR) algorithm. We first introduce the visual servoing approach and the application of the task function concept to vision-based tasks. Then, we present a complete control scheme which explicitly enables to pursue a moving object. In order to make the tracking errors as low as possible, we use the GLR test, an algorithm able to detect, estimate and compensate abrupt jumps in target motion. Finally, real-time experimental results using a camera mounted on the end effector of a six-d.o.f. robot are presented.

1 Introduction

Visual servoing [5] [6] [8] [10] is now a classical approach to realize various robotics tasks (positioning, grasping, target tracking, etc) in closed loop with respect to visual data.

As far as target tracking is concerned, Papanikolopoulos *et al.* [8] use classical approaches in control theory to track a moving object. However, they consider the object motion as disturbance, which implies tracking errors in the image. On the other hand, Allen *et al.* [1] have developed an object motion estimation algorithm, based on $\alpha - \beta - \gamma$ filters, in order to reduce the observed tracking errors. Other similar techniques are based on the use of Kalman filters [4], [7]. While the first approach has computational advantages, the second one seems much more appealing, thanks to the adaptability of its coefficients for tracking various target motions.

In this paper, we are also interested in target tracking. More precisely, we present a control scheme able to minimize the tracking errors due to the target motion. Our method is based on the task function approach [9], which has been applied to visual servoing in [5]. We use in this paper an algorithm able to detect, estimate and compensate abrupt changes in the target motion. This method is based on the Generalized Likelihood Ration test (GLR) developed by Willsky [11]. The experimental results described at the end of this paper show the robustness of the proposed control scheme with respect to measurement errors and unknown target motion.

2 Visual Servoing

In visual servoing, vision data are modeled as a set $\underline{s}$ of elementary visual signals which only depends on the relative position and orientation between the camera and the scene. For a given vision-based task, a desired target image is built, consisting of a chosen set of values $\underline{s}^*$, and the control problem is thus reduced to the regulation in the image of $(\underline{s} - \underline{s}^*)$.

Referring to earlier developments [5], the time variation of $\underline{s}$ can be modeled through:

$$\dot{\underline{s}} \;=\; L^T_{\underline{s}} \; T \;, \tag{1}$$

where

- $T = (V, \Omega)$ is the velocity screw quantifying the relative motion between the considered target and the camera;
- $L^T_{\underline{s}}$, called the interaction matrix related to $\underline{s}$, completely characterizes the interaction between the sensor and the target.

Let us now consider the situation $\underline{r}$ of the camera with respect to a reference frame. We have $\underline{s} = \underline{s}(\underline{r}, t)$ where the time variable t denotes the target motion. Applying the task function approach [9] to the case of visual sensor allows us to define a vision-based task function $\underline{e}$ of the form [5]:

$$\underline{e} \;= \widehat{L^T_{\underline{s}}}^{+} \; (\underline{s}(\underline{r}, t) \;-\; \underline{s}^*) \;\;, \tag{2}$$

where $\widehat{L^T_{\underline{s}}}^{+}$ is the pseudo-inverse of a chosen model of $L^T_{\underline{s}}$. Considering the control problem as a closed loop regulation of $\underline{e}$, we can ensure that the task is perfectly achieved if, at each time t, $\underline{e}(\underline{r(t)}, t) \;=\; 0$. In order that $\underline{e}$ exponentially decreases, the desired evolution of $\underline{e}$ takes the form:

$$\dot{\underline{e}} = -\lambda \; \underline{e} \;, \tag{3}$$

where λ (> 0) controls the speed of the decay. Since $\underline{e}$ is function of $\underline{r}$ and t, we have:

$$\dot{\underline{e}} \;=\; \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \; T_c \;+\; \frac{\partial \underline{e}}{\partial t} \tag{4}$$

where $T_c = \frac{\partial \underline{r}}{\partial t}$ is the camera velocity and where $\frac{\partial \underline{e}}{\partial t}$ represents the variation of $\underline{e}$ due to the target motion. We thus obtain, from (3) and (4):

$$T_c \; = \; \left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}}\right)^{+} \left(-\lambda \; \underline{e} \; - \; \widehat{\frac{\partial \underline{e}}{\partial t}}\right) \; , \qquad (5)$$

where

- $\left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}}\right)$ can be taken as the identity matrix under certain conditions described in [9], particularly for our vision-based task ; and
- $\widehat{\frac{\partial \underline{e}}{\partial t}}$, on which we will now focus, is the estimate of the target motion.

Using equation (4), we can easily obtain a measure of the target motion in the image:

$$\left(\widehat{\frac{\partial \underline{e}}{\partial t}}\right) = \widehat{\dot{\underline{e}}} - \left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}}\right) T_c \; . \qquad (6)$$

After discretization, this relation becomes:

$$\left(\widehat{\frac{\partial \underline{e}}{\partial t}}\right)_{(k)} = \frac{\underline{e}_{(k)} - \underline{e}_{(k-1)}}{\Delta t} - \left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}}\right)_{(k)} T_{c(k-1)} \qquad (7)$$

where Δt is the sampling period of the control law.

3 Filtering Target Motion

Two different sources for noise are possible in our estimation scheme: it can be either introduced through the extraction of the visual data or due to robot joint position measurement errors. In order to obtain a robust estimation of the target motion, we have chosen to use Kalman filtering. We use constant velocity or acceleration models with colored noise [4] [7] instead of white noise. This colored noise enables to consider low variations in the state model.

Furthermore, we use in parallel the GLR algorithm [11]. The obtained behavior should be exactly the same as the Kalman filter as long as the target does not maneuver. When this happens, the GLR test detects and estimates the jump in the target motion, which is used to update the estimate of the Kalman filter. This allows us to obtain an accurate estimate of the state vector with a shorter delay than using a classical Kalman filter. Indeed, the Kalman filter considers a jump as noise and needs several iterations before yielding a correct estimate. In the remainder of this Section, we first recall the general GLR method, and then apply it to two particular cases: the detection of velocity jump [3] and acceleration jump.

3.1 General case

The state and observation models used in the Kalman filter (with a jump in state vector at time θ) are respectively:

$$x_{(k+1)} = \Phi \; x_{(k)} + v_{(k)} + \; \delta_{\theta,k+1} \; \nu \qquad (8)$$

$$z_{(k+1)} = H \; x_{(k+1)} + \; w_{(k)} \qquad (9)$$

where v is the zero-mean gaussian white noise on the state model, δ_{ij} is the Kronecker symbol, ν is the hypothetic jump, and w is the zero-mean gaussian white noise on the observation.

We want to detect any jump in the target motion that occurs at an unknown instant θ. When a jump occurs at time θ, this has an effect on the value of the measurement residual $\gamma(k)$ of covariance $V(k)$ and therefore on the state estimate. $\gamma(k)$ can thus be considered as a sum of two terms:

$$\gamma_{(k)} = \gamma_{(k)_{nj}} + G(k;\theta) \; \nu \qquad (10)$$

where $\gamma_{(k)_{nj}}$ represents the measurement residual if a jump does not occur, and $G(k;\theta) \; \nu$ is the effect of a jump, which occurred at iteration θ, on $\gamma(k)$ measured at iteration k. Similarly, the state vector can be written as the sum of two quantities:

$$x_{(k|k)} = x_{(k|k)\,nj} + F(k;\theta) \; \nu \qquad (11)$$

As shown in [11], the detection and the estimation of a jump are based on the computation of the matrices $G(k;\theta)$ and $F(k;\theta)$, which can be recursively obtained from $G(k-1;\theta)$ and $F(k-1;\theta)$, computed at the previous iteration. The detection of a possible jump occurring at time θ is based on the value of the likelihood ratio $l(k;\theta)$ given by the following equation [11]:

$$l(k;\theta) = D^T(k;\theta) \; C^{-1}(k;\theta) \; D(k;\theta) \qquad (12)$$

with:

$$C(k;\theta) = \sum_{j=\theta}^{k} G^T(j;\theta) \; V_{(j)}^{-1} \; G(j;\theta) \qquad (13)$$

and:

$$D(k;\theta) = \sum_{j=\theta}^{k} G^T(j;\theta) \; V_{(j)}^{-1} \; \gamma_{(j)} \qquad (14)$$

$l(k;\theta)$ measures the correlation between the variations of the measurement residual $\gamma_{(k)}$ and the signature $G(k,\theta)$ of any jump. We compute the ratio $l(k;\theta)$ for a range of θ values, and we select the value θ_m which maximizes the quantity $l(k;\theta)$ and represents the most probable time at which a jump occurred. If a jump is detected $(l(k;\theta_m) > \epsilon$, where ϵ is a chosen threshold value), the iteration $\hat{\theta}$ of its occurring and the estimation of its value $\hat{\nu}(k;\hat{\theta})$ are given by:

$$\begin{cases} \hat{\theta} \; = \; \theta_m \\ \hat{\nu}(k;\hat{\theta}) = C^{-1}(k;\hat{\theta}) \; D(k;\hat{\theta}) \end{cases}$$

The estimation of the jump is then used to update the state vector estimate of the Kalman filter by the following compensation equation [11]:

$$x_{(k|k)new} = x_{(k|k)old} \; + \; [\Phi^{k-\hat{\theta}} \; - \; F(k;\hat{\theta})] \; \hat{\nu}(k;\hat{\theta})$$

Finally, in order to take into account the error in the jump estimate, we increase the error covariance update matrix $P_{(k|k)}$. This operation allows us to avoid false alarms after the detection of a jump.

$$\begin{aligned} P_{(k|k)new} &= P_{(k|k)old} \\ &+ [\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})] \, C^{-1}(k;\hat{\theta}) \, [\Phi^{k-\hat{\theta}} \; - \; F(k;\hat{\theta})]^T \end{aligned}$$

3.2 Constant velocity Kalman filter, jump in velocity

We now apply the GLR algorithm to our particular case of target tracking by visual servoing. We want to detect, estimate and compensate abrupt changes in target velocity. Since there is no correlation between the six different components of the target motion, we have implemented a such approach for each component $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ of $\left(\widehat{\frac{\partial e}{\partial t}}\right)$.

We choose a constant velocity state model with colored noise. In this particular case, the state and observation equations of the Kalman filter are given by:

$$\begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \end{pmatrix} = \Phi \begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k)} \\ \eta_{(k)} \end{pmatrix} + \begin{pmatrix} 0 \\ v_{(k)} \end{pmatrix} + \; \delta_{\theta,k+1} \begin{pmatrix} \alpha \\ 0 \end{pmatrix} \quad (15)$$

$$\left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k+1)} = H \begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \end{pmatrix} + \; w_{(k+1)} \quad (16)$$

with: $\qquad \Phi = \begin{pmatrix} 1 & 1 \\ 0 & \rho \end{pmatrix}$, $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$

and where α is the magnitude of a hypothetic jump in velocity, and $\left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k)}$ is the observed target velocity at time k (given by equation (7)).

We consider a Kalman filter with the model described above, and search to detect jumps of the target velocity. Since the direction of the jump in the state space is $a = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$, the measurement residual is given by:

$$\gamma_{(k)} = \gamma_{(k)_{nj}} + g(k;\theta) \; \alpha \quad (17)$$

where $g(k;\theta)$ is a scalar. Similarly, the state vector is written as a sum of two terms:

$$\begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix} = \begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}_{nj} + \alpha \; f(k;\theta) \quad (18)$$

$g(k;\theta)$ and $f(k;\theta)$ are respectively the result of multiplying $G(k;\theta)$ and $F(k;\theta)$ by the direction of the jump a.

We obtain [2] : for $\theta = k$

$$\left\{ \begin{array}{l} g(\theta;\theta) = G(\theta;\theta) \; a = 1 \\ f(\theta;\theta) = F(\theta;\theta) \; a = K_{(\theta)} \end{array} \right.$$

where K is the Kalman filter gain, and for $\theta < k$:

$$\left\{ \begin{array}{l} g(k;\theta) = G(k;\theta) \; a = 1 - \begin{pmatrix} 1 & 0 \end{pmatrix} \; f(k-1;\theta) \\ f(k;\theta) = F(k;\theta) \; a = \Phi \; f(k-1;\theta) + g(k;\theta) \; K_{(k)} \end{array} \right.$$

In this case, the likelihood ratio defined by the equation (12) can be written as follows:

$$l(k;\theta) = \frac{d^2(k;\theta)}{c(k;\theta)}$$

with (see equation (13)): $c(k;\theta) = \sum_{j=\theta}^{k} \frac{g^2(j;\theta)}{v_{(j)}}$

and (see equation (14)): $d(k;\theta) = \sum_{j=\theta}^{k} \frac{g(j;\theta) \, \gamma_{(j)}}{v_{(j)}}$

If a jump is detected ($l(k;\theta_m) > \epsilon$ where ϵ is a chosen threshold value, and where θ_m is the value which maximizes $l(k;\theta)$), its occurring time estimate is given by $\hat{\theta} \; = \; \theta_m$ and its magnitude by:

$$\hat{\alpha}(k;\hat{\theta}) = \frac{d(k;\hat{\theta})}{c(k;\hat{\theta})}$$

Finally, the compensation equation used in the Kalman filter to update the state vector is:

$$\begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}_{new} = \begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}_{old} + \begin{pmatrix} \hat{\alpha}(k;\hat{\theta}) \\ 0 \end{pmatrix} - \hat{\alpha}(k;\hat{\theta}) \; f(k;\hat{\theta})$$

The compensation is done by adding the jump estimate in the state space to the latest state vector estimate, and by substracting the response of the Kalman filter to the jump before its detection.

3.3 Constant acceleration Kalman filter, jump in acceleration

In this Section, we apply the GLR algorithm to another particular case. We are here interested in detecting jumps in acceleration for a state model based on constant acceleration and colored noise. The new state and observation equations of the Kalman filter are given by:

$$\begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \\ \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \end{pmatrix} = \Phi \begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k)} \\ \eta_{(k)} \\ \left(\frac{\partial e}{\partial t}\right)_{(k)} \end{pmatrix} + \begin{pmatrix} 0 \\ v_{1(k)} \\ v_{2(k)} \end{pmatrix}$$

$$+ \; \delta_{\theta,k+1} \begin{pmatrix} 0 \\ 0 \\ \alpha \end{pmatrix} \quad (19)$$

and:

$$\left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k+1)} = H \begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \\ \left(\frac{\dot{\partial} e}{\partial t}\right)_{(k+1)} \end{pmatrix} + w_{(k+1)} \qquad (20)$$

with: $\Phi = \begin{pmatrix} 1 & 1 & \Delta t \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$

The direction of the jump in the state space is now $a = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T$. In this case, we obtain [2]:

$$\begin{cases} g(\theta;\theta) = G(\theta;\theta)\ a = 0 \\ f(\theta;\theta) = F(\theta;\theta)\ a = (0) \end{cases}$$

This result was expected. Indeed, if a jump in acceleration occurs at time θ, its effect on the velocity, which is the only observed variable, begins at time $\theta+1$. On the other hand, we have:

$$\begin{cases} g(k;\theta) = G(k;\theta)\ a = (k-\theta)\ \Delta t - \begin{pmatrix} 1 & 1 & \Delta t \end{pmatrix}\ f(k-1;\theta) \\ f(k;\theta) = F(k;\theta)\ a = \Phi\ f(k-1;\theta) + g(k;\theta)\ K_{(k)} \end{cases}$$

The computation of the likelihood ratio, and the detection and estimation of a hypothetic jump are done as in the previous section. The compensation equation used in the Kalman filter to update the state vector is now:

$$\begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \\ \left(\widehat{\frac{\dot{\partial} e}{\partial t}}\right)_{(k|k)} \end{pmatrix}_{new} = \begin{pmatrix} \left(\widehat{\frac{\partial e}{\partial t}}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \\ \left(\widehat{\frac{\dot{\partial} e}{\partial t}}\right)_{(k|k)} \end{pmatrix}_{old}$$

$$+ \begin{pmatrix} (k-\theta)\ \Delta t\ \hat{\alpha}(k;\hat{\theta}) \\ 0 \\ \hat{\alpha}(k;\hat{\theta}) \end{pmatrix} - \hat{\alpha}(k;\hat{\theta})\ f(k;\hat{\theta})$$

The compensation is here done by adding the effect of a jump in acceleration of magnitude $\hat{\alpha}(k;\hat{\theta})$ on the state vector during $(k-\theta)$ iterations, and by substracting the response of the Kalman filter to the jump before its detection.

4 Experimental Results

For our experiments, we used a CCD camera mounted on the end-effector of a six d.o.f. robot. In order to demonstrate the generality of our approach, two different target tracking tasks have been implemented:

- the first one consists in controlling the six camera d.o.f. in order to realize a rigid link between the camera and the target.
- the second one consists in controlling the camera orientation in order that the image of the target appears at the center of the image plane.

For both tasks, the target tracking has been realized using the four following methods:

- without estimating the target motion. In that case, tracking errors should appear in the image during the target motion.
- using the estimation of the target motion obtained with a Kalman filter with constant velocity and colored noise state model;
- using the estimation of the target motion obtained with a Kalman filter with constant acceleration and colored noise state model. This filter should respond more rapidly to velocity changes. However, it should also be more sensitive to measurement noises.
- using the GLR algorithm to detect and compensate abrupt changes in the velocity or the acceleration of the target motion.

4.1 Tracking a square

This first target tracking task consists in controlling the camera position and orientation such that the image of a square always remains at the same position in the image. More precisely, the visual data used in the vision-based task are the coordinates of four points representing the square corners (see Figure 1.a). The image corresponding to the chosen location of the camera with respect to the square is given in Figure 1.b.

The experiment consists of the following steps: initially, the square is motionless and the camera reaches its desired position using visual servoing. Then, the square begins a translational motion with a constant velocity of 5 cm/s along 90 cm. After a stop, it moves back to its original departure position with the same velocity.

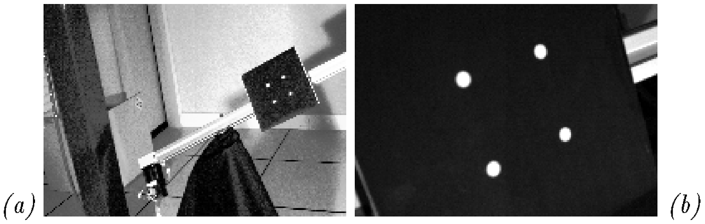
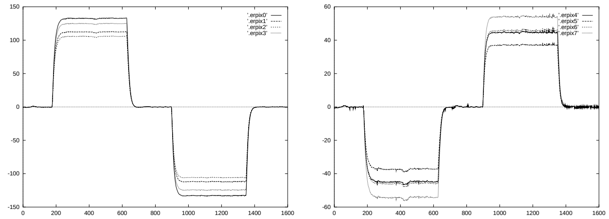


Figure 1: Images acquired at the initial and final camera positions

4.1.1 Results without estimating the target motion

The results obtained when the square motion is not estimated are depicted on Figure 2. More precisely, Figure 2.a (resp. Figure 2.b) represents, at each iteration of the control law, the measured value in the image of the first four components of $(\underline{s} - \underline{s}^*)$ (resp. the last four components).

When the object is motionless, we can observe a maximum error in the image of about 1 pixel. These small perturbations are due to image measurement errors. During the square motion, the error observed in the image implies a tracking camera motion. However, the control law is not able to compensate these errors, since the assumption $\widehat{\frac{\partial e}{\partial t}} = 0$ produces an important tracking error (which is constant since the object has a constant velocity) of about 135 and 55 pixels on the horizontal and vertical image axes respectively.



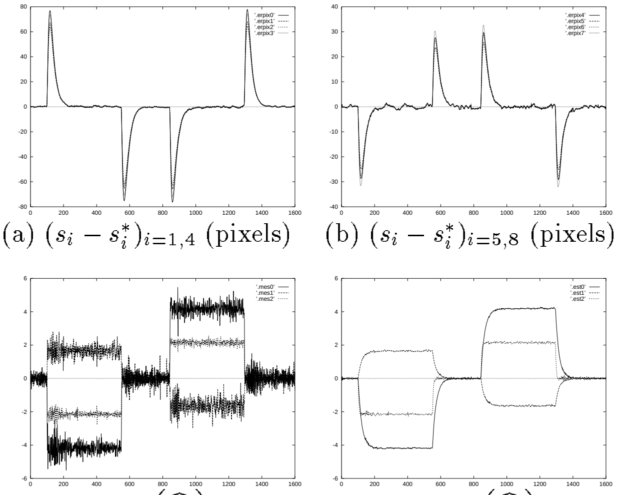
(a) $(s_i - s_i^*)_{i=1,4}$ (pixels) (b) $(s_i - s_i^*)_{i=5,8}$ (pixels)

Figure 2: Square tracking without motion estimation

4.1.2 Kalman filter with constant velocity and colored noise model

When the target is motionless, we can observe the same errors than in the previous case, which means that the Kalman filter does not perturb the behavior of the control law. When the square moves with a constant velocity, the tracking errors are now suppressed, since we obtain 1.5 pixel as maximum error in the image (see Figure 3.a and 3.b). The square motion is accurately estimated.

However, at the beginning and the end of the square motion, we obtain a delay in estimating an accurate value of the target velocity. This has an important effect on the error in the image (maximum error of 80 and 30 pixels on the X and Y axes respectively). Furthermore, 120 iterations are necessary to obtain an error in the image less than 1 pixel.



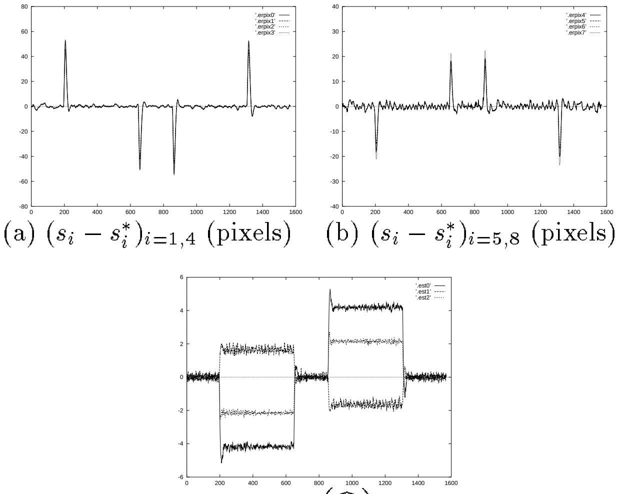
(a) $(s_i - s_i^*)_{i=1,4}$ (pixels) (b) $(s_i - s_i^*)_{i=5,8}$ (pixels)

(c) measured $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ (cm/s) (d) estimated $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ (cm/s)

Figure 3: Square tracking using a Kalman filter with constant velocity and colored noise state model

4.1.3 Kalman filter with constant acceleration and colored noise model

When the square has a constant velocity, we obtain a maximum error of 2 pixels in the image. This value, more important than in the previous cases, show that this model is not robust to measurement noises, which implies an unstable control law. However, there is no delay in estimating the new value of the target velocity when abrupt changes occur. Indeed, it only takes 55 iterations, including an oscillatory behavior, before the establishment of an error equivalent to the one recorded when the square is motionless. Furthermore, the maximum error observed in the image is equal to 55 and 25 pixels on the two image plane axes, which means that the jump is immediately taken into account.



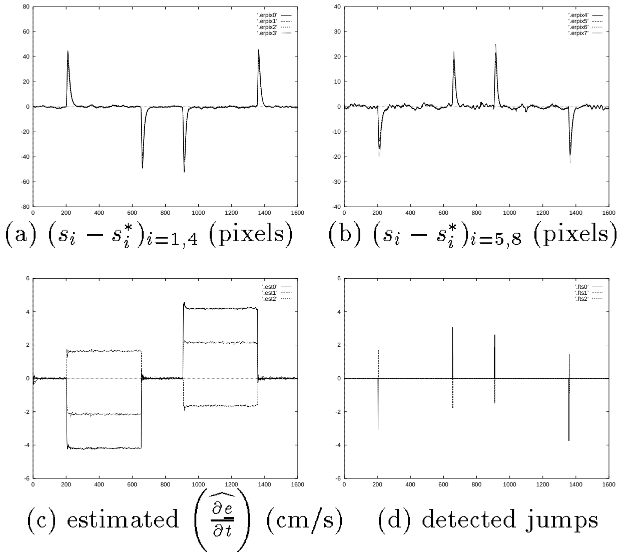
(a) $(s_i - s_i^*)_{i=1,4}$ (pixels) (b) $(s_i - s_i^*)_{i=5,8}$ (pixels)

(c) estimated $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ (cm/s)

Figure 4: Square tracking using a Kalman filter with constant acceleration and colored noise state model

4.1.4 GLR test and constant velocity Kalman filter model

Figure 5.d shows the iterations where jumps have been detected and their magnitude. This method has the advantages of the two precedent ones and not their respective disadvantages. Indeed, it correctly detects and estimates the jumps in target velocity while ensuring a robust and stable camera motion. When the target is motionless, its behavior is exactly the same as the Kalman filter with constant velocity state model since no false detection are performed. When the square begins or ends a motion, we obtain a highest error of 45 and 25 pixels on the image axes, and only 50 iterations are necessary before recording again a maximum error of 1 pixel. The performances are thus better than those obtained with the Kalman filter based on a constant acceleration state model.



(a) $(s_i - s_i^*)_{i=1,4}$ (pixels) (b) $(s_i - s_i^*)_{i=5,8}$ (pixels)

(c) estimated $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ (cm/s) (d) detected jumps

Figure 5: Square tracking using the GLR algorithm

4.2 Target Tracking using Pan and Tilt

For this second target tracking task, we use two camera d.o.f., i.e., its pan and tilt, in order that the center of gravity of the target in the image plane appears at the image center.

This experiment, which allows us to test the behavior of our algorithms with respect to jumps in the target accelerations, consists of the following steps: first, the control law (5) is used to orient the camera such that the motionless point appears at the center of the image (see Figure 6). Then, the translational robot d.o.f performs a dedicated trajectory (composed of successive accelerations and decelerations along the X camera axis, see Figure 7.a) in order to simulate a similar motion of the target. Indeed, this translational motion must be compensated by rotational motion of the camera in order that the target keeps its specified position in the image.

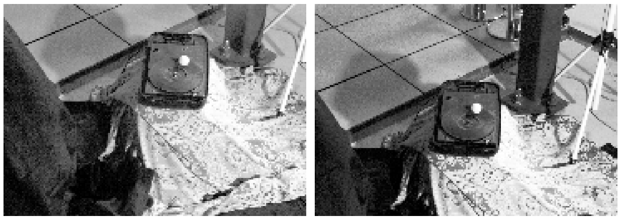
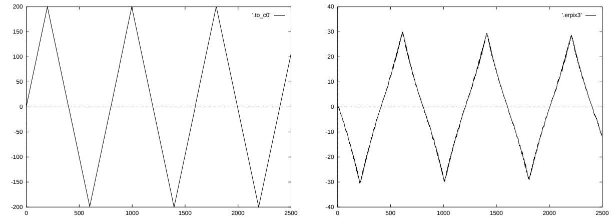


Figure 6: Initial and final images acquired by the camera

4.2.1 Results without estimating the target motion

Important tracking errors, whose maximum value is about 30 pixels, appear in the image (see Figure 7.b). We can note that the errors follow the target motion model.

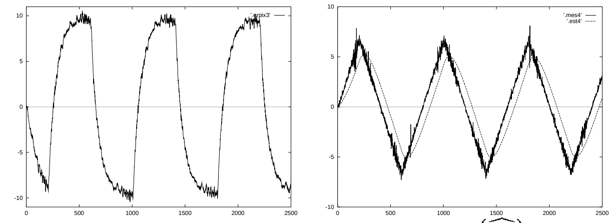


(a) Target motion (b) $\underline{s} - \underline{s}^*$ (pixels)

Figure 7: Target tracking without motion estimation

4.2.2 Kalman filter with constant velocity and colored noise model

In that case, the motion model involved in the Kalman filter does not correspond to the real one. That is why we obtain a delay of about 70 iterations in estimating the target velocity (see Figure 8.b). This delay induces permanent tracking errors, the maximum value is about 10 pixels (see Figure 8.a).

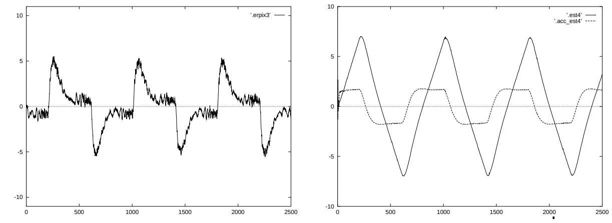


(a) $\underline{s} - \underline{s}^*$ (pixels) (b) meas. and est. $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ (dg/s)

Figure 8: Target tracking using a constant velocity and colored noise Kalman filter

4.2.3 Kalman filter with constant acceleration and colored noise model

In that case, the motion model involved in the Kalman filter corresponds to the one of the real target motion. This implies the suppression of the errors in the image, except during abrupt changes of the object acceleration (see Figure 9.a). During these changes, the object velocity is not accurately estimated since an important delay is necessary to obtain a correct estimation of the new target acceleration (see Figure 9.b). The maximum error in the image is now 5.5 pixels. Let us note that, because of the delay, an error in the image of about 1 pixel is reached after 150 iterations of the control law.

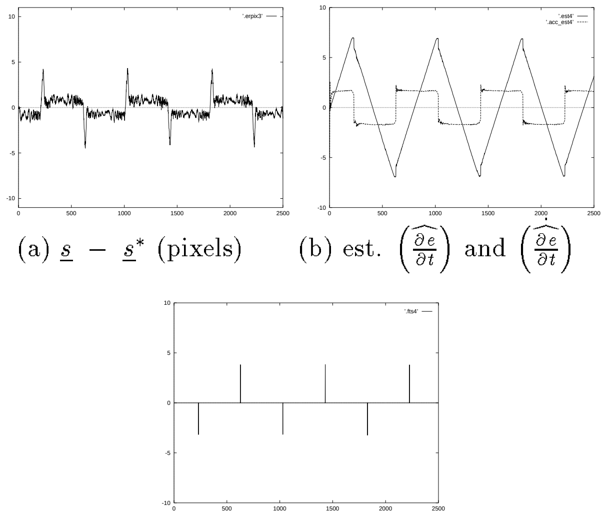


(a) $\underline{s} - \underline{s}^*$ (pixels) (b) est. $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ and $\left(\widehat{\dot{\frac{\partial e}{\partial t}}}\right)$

Figure 9: Target tracking using a constant acceleration and colored noise Kalman filter

4.2.4 GLR test and constant acceleration Kalman filter model

In that last case, the acceleration jumps are detected (see Figure 10.c) thanks to the use of the GLR test. This algorithm enables to suppress the delay in the estimation of the object acceleration, which is immediately taken into account by the estimation of the object velocity (see Figure 10.b). We now have a maximum error in the image equal to 4.5 pixels and only 30 iterations are necessary to obtain an error in the image less than 1 pixel.



(a) $\underline{s} - \underline{s}^*$ (pixels) (b) est. $\left(\widehat{\frac{\partial e}{\partial t}}\right)$ and $\left(\dot{\widehat{\frac{\partial e}{\partial t}}}\right)$

(c) detected jumps

Figure 10: Target tracking using the GLR test

We have also obtained satisfactory results using the GLR test when the target has a complex motion, such as combinations of rotations and translations with various accelerations, decelerations and velocities. As reported in the experiments described above, the GLR algorithm enables to significantly reduce the error in the image.

5 Conclusion

We have presented in this paper a visual servoing scheme using the task function approach. This scheme specifies the control problem in terms of a regulation in the image using 2D visual data. We have proposed a new control law based on this approach which enables to track an object with unknown motion. For doing that, a robust estimation and prediction scheme of the target motion in the image has been presented and introduced in the control law. Experimental results outline the fact that the target motion estimation through Kalman filtering combined with a GLR test is able to detect and compensate abrupt changes in the target motion.

References

[1] P.K. Allen, A. Timcenko, B. Yoshimi, P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system", *IEEE Trans. on Robotics and Automation*, Vol. 9, n. 2:pp. 152–165, April 1993.

[2] F. Bensalah, F. Chaumette, "Détection de rupture de modèle appliquée à l'asservissement visuel", *INRIA Research Report, n° 2425*, Rennes, November 1994.

[3] F. Bensalah, F. Chaumette, "Real time Visual Tracking using the Generalized Likelihood Ratio test", *Proc. of* 3^{th} *Int. Conf. on Automation, Robotics and Computer Vision*, Vol. 2:pp. 1379–1383, Singapore, November 1994.

[4] F. Chaumette, A. Santos, "Tracking a moving object by visual servoing", *Proc. of* 12^{th} *World Congress IFAC*, Vol. 9:pp. 409–414, Sydney, July 1993.

[5] B. Espiau, F. Chaumette, P. Rives, "A new approach to visual servoing in robotics", *IEEE Trans. on Robotics and Automation*, Vol. 8, n. 3:pp. 313–326, June 1992.

[6] K. Hashimoto, editor, *Visual servoing*, World Scientific Series in Robotics and Automated Systems, Vol. 7, World Scientific, Singapore, 1993.

[7] A.E. Hunt, A.C. Sanderson, "Vision-based predictive robotic tracking of a moving object", *Carnegie-Mellon University Technical Report, n° 82.15*, January 1982.

[8] N.P. Papanikolopoulos, P.K. Khosla, T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision", *IEEE Trans. on Robotics and Automation*, Vol. 9, n. 1:pp. 14–35, February 1993.

[9] C. Samson, B. Espiau, M. Le Borgne, *Robot control: the task function approach*, Clarendon Press, Oxford, England, 1991.

[10] L.E. Weiss, A.C. Sanderson, C.P. Neuman, "Dynamic sensor-based control of robots with visual feedback", *IEEE Journal of Robotics and Automation*, Vol. 3, n. 5:pp. 404–417, October 1987.

[11] A.S. Willsky, H.L. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems", *IEEE Transactions on Automatic Control*, Vol. 21, n. 1:pp. 108–112, February 1976.