

# Image-based visual servoing by integration of dynamic measurements

Armel Crétual   François Chaumette  
IRISA / INRIA Rennes  
Campus de Beaulieu  
35042 Rennes cedex, France  
E-mail {acretual, chaumett}@irisa.fr

## Abstract

*Visual servoing based upon geometrical features such as image points coordinates is now well set on. Nevertheless, this approach has the drawback that it usually needs visual marks on the observed object to retrieve geometric features. The idea developed here is that these features can be retrieved by integrating dynamic ones, which can be estimated without any a priori knowledge of the scene. Thus, more realistic objects can be used to achieve vision-based control such as tracking and fixation tasks. We detail control laws concerning these two tasks, first using integration of speed in the image and then by direct regulation of these dynamic parameters. Results are finally presented and comparisons are made between the two types of control methods.*

## 1 Introduction

The aim of visual servoing, as presented in [11, 13], is to control the robot displacements using visual features. One of the method used to complete such control laws is to apply the task function approach [19] to visual sensors and is based on the linear relation existing between image features variation and camera motion [9]. Geometric primitives have most often been used until now to complete robotic tasks such as positioning with respect to a given object. For example, visual marks are used in [9] to extract geometric features from the image, whereas [5, 7, 10] use informations from the object contour or particular features such as corners. Convergence is usually ensured and stable, at least when the initial position is in the 3D neighborhood of the desired position, and most of these applications run at video rate. The major problem encountered is that an a priori knowledge of the geometric features is needed.

Visual servoing based on dynamic features has recently been developed [8, 18, 21]. In this case, no knowledge of the observed pattern is necessary. The useful informations are extracted from 2D motion be-

tween two successive images. It can be seen that tasks such as tracking a moving object can be solved by this approach. To maintain the object at the same position in the image is equivalent to make its 2D projection keep a null speed.

Several papers deal with target tracking of mobile object. [3, 6, 12, 17] use visual marks to extract geometric information which are reinforced by an estimation of object speed in the image to compensate errors. Another method is used in [2, 15], but it is only able to track a small object. An affine model of 2D motion is computed between two successive images and the second image is compensated with the opposite motion. Threshold difference between this new image and the first one gives the position of the object, and the camera is controlled in pan and tilt so that this position stays at the image center. Finally, [1] uses a stereo-vision system to build a 3D model of the object motion in order to position the robot arm to grasp it.

This paper proposes a new approach to regulate speed in the image to zero. The problem of using directly dynamic visual features in the control loop is that, as the order of derivation is increased by one, there is generally no more a linear relation between features variations and camera motion. Furthermore, drift due to reacting time are not compensated. For these reasons, this paper develop the idea that position in the image can be retrieved by integration along time of speed in the image. Thus, visual servoing, as it is done with geometric features, can be used, but visual marks are no longer necessary. The principle of servoing by retrieving position from speed is quickly exposed in Section 2. Two applications are then presented and compared to methods using directly dynamic visual features in the control loop. We first describe the corresponding control laws, and then, display results obtained on our eye in hand 6 d.o.f. robotic system. The first application, detailed in Section 3, is the tracking of a mobile object using camera pan and tilt. The other one, in Section 4, corresponds to the positioning of the camera parallel to a plane coupled to a fixation task.

## 2 Image-based control from speed measurements

Our aim is to control the robot by classical image-based techniques but without having any a priori knowledge on the image content. The solution proposed is to retrieve geometric features by integrating dynamic measurements along time.

Let us call  $s = (x, y)^T$ , the 2D projection at time  $t$  of a 3D point  $M$ , and  $\dot{s}$  its apparent speed in the image.  $s$  can obviously be retrieved knowing the projection  $s_0$  at time 0 and the evolution of  $\dot{s}$  along time, by:

$$s = s_0 + \int_0^t \dot{s} dt$$

This relation can be approximated under the following discrete form:

$$s = s_0 + \sum_{i=1}^k \dot{s}_i \delta t_i \quad (1)$$

with  $\dot{s}_i$  being the  $i^{\text{th}}$  measurement of  $\dot{s}$  and  $\delta t_i$ , the duration between  $(i-1)^{\text{th}}$  and  $i^{\text{th}}$  measurements.

The motion model used to approximate speed in the image is a simplified quadratic model with 8 parameters as below (see [8, 20]):

$$\begin{cases} \dot{x} &= a_1 + a_2x + a_3y + b_1x^2 + b_2xy \\ \dot{y} &= a_4 + a_5x + a_6y + b_3y^2 + b_4xy \end{cases} \quad (2)$$

with

$$\begin{cases} a_1 = -v_x - \Omega_y & a_2 = \gamma_1 v_x + v_z & a_3 = \gamma_2 v_x + \Omega_z \\ a_4 = -v_y + \Omega_x & a_5 = \gamma_1 v_y - \Omega_z & a_6 = \gamma_2 v_y + v_z \\ b_1 = -\gamma_1 v_z - \Omega_y & b_2 = -\gamma_2 v_z + \Omega_x & b_3 = b_2 \quad b_4 = b_1 \end{cases}$$

where  $T$  and  $\Omega$  respectively represent the translational and the rotational terms of the kinematic screw between the camera frame and the observed object frame,  $(v_x, v_y, v_z) = \frac{1}{Z_p} T$ , and  $Z = Z_p + \gamma_1 X + \gamma_2 Y$  is the equation of the planar approximation of the object surface at the considered point expressed in the camera frame. The algorithm used to estimate parameters  $a_i$  and  $b_i$  is the RMRm (robust multi-resolution algorithm) developed in [16].

Of course, simpler models (constant, affine) can be used to estimate the position of the image center, which are simply deduced from the one presented above by identification with the corresponding parameters. In fact, there is a necessary compromise to find between precision of the estimation and duration of calculation.

The following block-diagram sums up the whole paragraph where  $s$  and  $s^*$ , respectively represent the current and desired position in the image,  $\Omega$  the controlled rotation and  $a_i$  the dynamic visual features which give the 2D speed integrated to retrieve  $s$ .

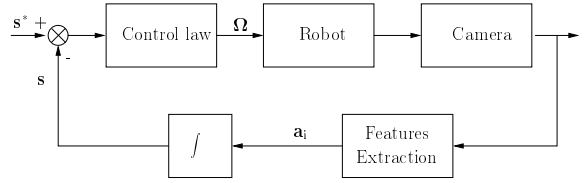


Figure 1: Block diagram: geometrical image-based control from dynamic visual features

## 3 Application to tracking

The principle of the tracking task is to control the camera pan and tilt such that a detected mobile object becomes projected at the center of the image or for want of anything better to keep it at the same position. We are not interested here in problems such as occlusions or multiple moving objects.

### 3.1 Detection of the mobile object

A step of detection of the mobile object has to be done first to obtain its initial projection mask on the image. As we do not use any a priori information on the target, this detection is performed using only the property that it is in motion. The camera remaining static until the mobile object is detected, the object projection location is determined by difference between two successive images. In practice, because of noise in the image, we use a local spatial ( $3 \times 3$  pixels) average of image intensities. Then, by considering a threshold difference between these two averaged images, we get a binary image separating moving zones from motionless ones. Under the hypothesis of a single moving object, it is easy to separate the mask of the mobile object from the background. The center of gravity of the mask gives the initial position which has to be regulated to zero, corresponding to  $s_0$  in (2). Once the detection is done,  $s$  is obtained by integration of speed parameters given by the RMRm algorithm.

### 3.2 Associated control law

Having the estimation of the center of gravity (c.o.g.) of the target from (1), we thus can use a standard control law to complete the regulation of this estimated 2D position.

Let us consider the vector of error  $s = (x, y)^T$ . Using the fact that the camera motions are only the rotations around the  $x$  and  $y$  axes, 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 2D motion of the target and  $\Omega_c$  the camera rotation.

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

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

The first term of this control law only allows to reach convergence when the observed object is motionless. To remove the tracking errors due to the object own motion, the second term has to be added and can be estimated by [4]:

$$\frac{\widehat{\partial s}}{\partial t} = \widehat{\dot{s}} - L\widehat{\Omega}_c$$

where the RMRm estimation algorithm gives  $\widehat{\dot{s}}$  and  $\widehat{\Omega}_c$  is the measured camera rotation. As done in [4], this estimation is filtered using a Kalman filter with a constant acceleration model with correlated noise.

### 3.3 Results

The tracking task has been tested on a 6 d.o.f. Cartesian robot cell, where the camera is mounted on the end-effector.  $256 \times 256$  images were acquired by a SunVideo Board and treated on an UltraSpark station. As we do not have any exact measurement of the c.o.g. when using a real complex object, the control law has been first tested with a simple target from which we can extract geometric features. The position of the c.o.g. is unused in the control law but can be compared to its estimation from the motion parameters. Thus, we can observe the exact behavior of the control law. The object was a black surface where 4 white circles formed a square (see initial image on Fig 2(a)). An image processing running at video rate gives the position of the c.o.g. for each circle. Displacements of this four centers and measurement of time between two successive images give the speed for each. Applying a affine model of motion, in order to get a sufficient precision on constant parameters, leads to a linear system of 6 unknowns with 8 equations, which is solved by least square method. Finally, the c.o.g. of the square is given by the intersection of its two diagonals. Then, the experiment has been made with a  $20 \times 20$  cm square from which no geometric features can be easily computed (see initial image on Fig 2(b)). In this case, to ensure a rate as closest as possible to the video rate, only the constant model of motion is used. With the object size such as it is obtained in this experiment, the rate reached is about 20 images per second.

The same conditions were taken for the two experiments, including initial positions of the target and of the camera. The target was translating along a rail

alternatively to the right and to the left at the same speed, with a 4 seconds pause between the two motions. Accelerations and decelerations were  $40 \text{ cm/s}^2$ . The camera was about 1 m away from the object which was seen before its first motion, but did not necessarily appear at the center of the image. The images were  $256 \times 256$  pixels.  $\lambda$  was chosen equal to 1.5. The successive motions are the following (where the number of iterations are approximative): it. 1 to 100, right move at 8 cm/s; it. 100 to 220, 4 seconds stop; it. 220 to 450, left move at 8 cm/s; it. 450 to 560, 4 seconds stop; it. 560 to 700, right move at 8 cm/s; it. 700 to 900, stop; it. 900 to 1000, right move at 30 cm/s; it. 1000 to 1100, 4 seconds stop, it. 1100 to 1220, left move at 30 cm/s; finally after iteration 1220, stop.

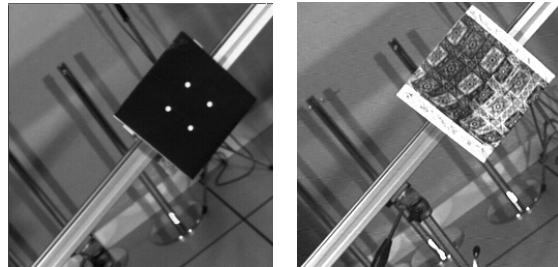


Figure 2: Initial images. (a) Four points object (b) "Real" square

For the first experiment, we present in Fig. 3 the difference between the estimated position of the center of the object and the measured one. This error is always less than 0.5 pixel. Thus, as there is no drift due to the integration, we conclude we can trust the estimation of the c.o.g.. Furthermore, previous experiments have been made, in order to compare noise in estimations of the constant parameters of motion by the two methods presented above, with known motions of the observed object. It showed that noise is not greater with the RMRm algorithm than with the four point estimation. It is even sometimes lower, in particular when the object is motionless.

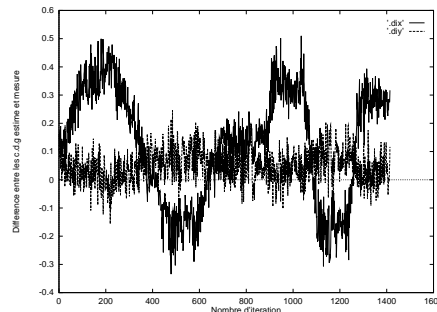


Figure 3: Four points experiment. Difference between the estimated displacement and the measured position of the center (in pixel)

The estimated displacement of the object center for the real square experiment, is displayed in Fig 4. This result is very similar to the one obtained with the four points experiment, and there is even less oscillations in the steps corresponding to the target stops. This experiment shows that convergence is well obtained for the initial error of about 40 pixels (it is brought to zero in less than 40 iterations even if the first motion of the object is on the opposite direction). At each abrupt change in the target motion (stop or start), there is an overrun due to the Kalman filter reacting time, but convergence is still obtained. This overrun of about 10 pixels is compensated in approximatively 20 iterations (0.8 seconds) when the speed is 8 cm/s (respectively 25 pixels and 50 iterations when the speed is 30 cm/s).

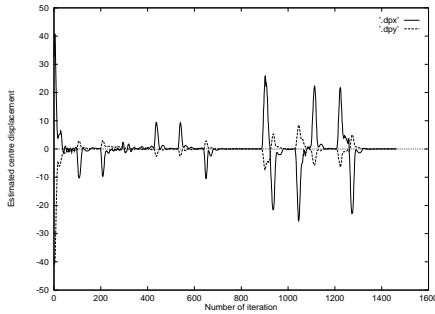


Figure 4: Square target experiment. Estimated displacement of the object center (in pixel)

The computed rotational velocities are displayed in (see Fig 5). A level of about 4 deg/s is necessary to track the 8 cm/s motion. In the case of the 30 cm/s motion, due to short time between start and stop of the object, we can note that  $\Omega_x$  and  $\Omega_y$  just reach their constant level of about 15 deg/s when the object stops, but no perturbation occurs in the regulation of the estimated c.o.g..

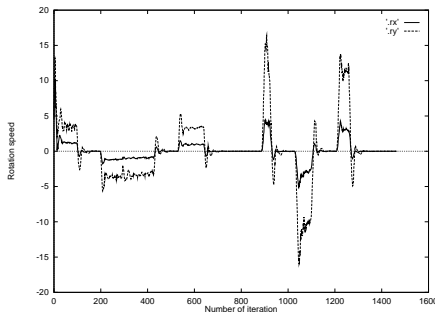


Figure 5: Square target experiment. Computed control law  $\Omega_x$  and  $\Omega_y$  (in deg/s)

### 3.4 Tracking with image motion based control

To see the interest of the previous method, a control directly based on image motion has been settled. The task consists now in trying to keep the object at the same position in the image by regulating to zero the constant terms of motion  $s_1 = (a_1, a_4)^T$ . Derivating  $s_1$  along time leads to the interaction relation [20]:

$$\begin{pmatrix} \dot{a}_1 \\ \dot{a}_4 \end{pmatrix} = L \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} + \frac{\partial s_1}{\partial t} \text{ with } L = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Applying the gradient type control, using a decreasing gain  $\lambda$ , the control law, expressed as angular acceleration, is then:

$$\begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} = -\lambda \begin{pmatrix} a_4 \\ -a_1 \end{pmatrix} - L^{-1} \frac{\partial \widehat{s}_1}{\partial t}$$

$\frac{\partial \widehat{s}_1}{\partial t}$  is estimated by  $\frac{\widehat{s}_{1k} - \widehat{s}_{1k-1}}{\delta t}$  where indices  $k$  and  $k - 1$  stand for current and previous values and  $\widehat{s}_1 = (a_1 + \Omega_{c,y}, a_4 - \Omega_{c,x})^T$  which is the zero order of  $\frac{\partial \widehat{s}}{\partial t}$  considered previously. Thus, it can be estimated with the same Kalman filter, only replacing  $\widehat{s}$  by measure of  $(\widehat{a}_1, \widehat{a}_4)^T$ . Direct filtering of  $\dot{s}_1$  is not accurate as the acceleration step is generally too short in front of the Kalman filter reacting time.

In the following summarizing block-diagram,  $s$  and  $s^*$  stand respectively for the current and desired values of the dynamic features  $(a_1, a_4)^T$ , which are given by the 2D-motion estimation algorithm and  $\dot{\Omega}$  is the controlled rotational acceleration.

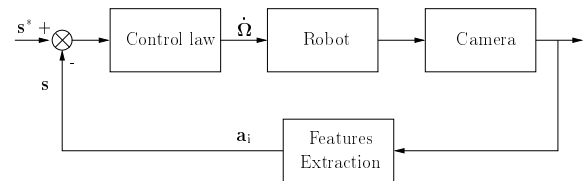


Figure 6: Block diagram: direct control of dynamic visual features

Let us note that in that case, the object can not be brought to the center of the image as we do not have anymore information on its position. Indeed, even if the detection step gives the first position of the c.o.g., trying to regulate it without any further information by an open loop would not be robust. Furthermore, if in the same time, the c.o.g. is still estimated and regulated to zero by rotational velocity, as regulation of  $s$  to zero is done by rotational acceleration, it would raise the problem, as in [14] of specifying the behavior of the control law by two different (and generally incompatible) means.

### 3.5 Results

The previous control law has been tested with the same initial condition and the same “real” target than in the previous results section. There, translational speeds were always 30 cm/s. First are presented the constant parameters of motion i.e.  $a_1$  and  $a_4$  (see Fig 7), the rotation acceleration computed by the control law (see Fig 8) and the estimated displacement of the object center (see Fig 9). The angular acceleration control law allows to fulfill the desired task which was to brought  $a_1$  and  $a_4$  to zero. It is accomplished in about 100 iterations after each abrupt change of motion, and it remains stable during the permanent running (during either a constant non-null speed motion or a motionless step), even if the computed parameters of control law are quite noisy. Finally, when the target becomes static again (between iterations 250 and 320, and between iterations 580 to 650), it appears a drift with respect to the initial position (1 or 2 pixels in the first stop, and about 10 to 15 in the second one). It is due to errors in the estimation of the object own acceleration. The accelerating step is very short (never more than 15 iterations), so the Kalman filter is not able to refine the estimated acceleration, and thus errors are not compensated.

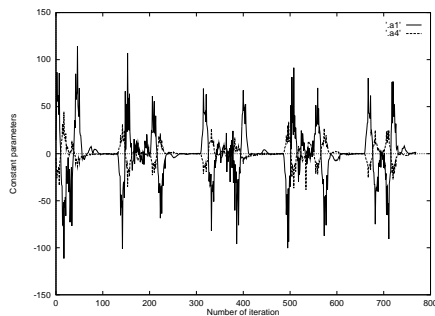


Figure 7: Square target experiment. Constant parameters of motion (in pixel/s)

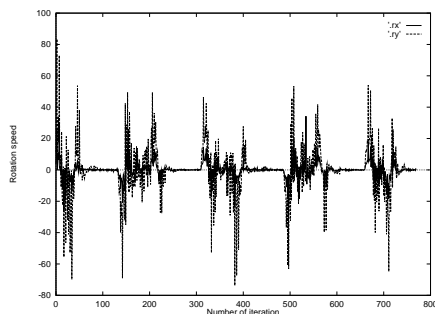


Figure 8: Square target experiment. Angular acceleration (in  $\text{deg/s}^2$ )

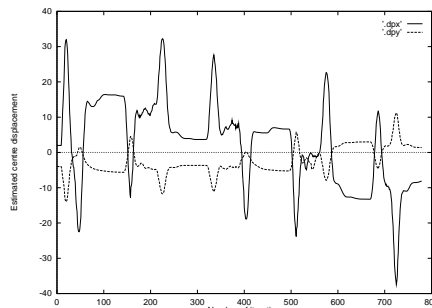


Figure 9: Square target experiment. Estimated displacement of the object center

### 3.6 Conclusion

The results presented here proves that tracking a real object, meaning without any visual marks, with an eye in hand system by visual servoing is possible whatever could its relative size be. It is solved by retrieving position of the object center by integration of its speed, and it runs close to video rate. On the contrary, dynamic control directly based on motion is unable to avoid a sensible drift of the object in the image. Furthermore, in the second case, the task aim was to keep the object in the same position. In most cases, a mobile object will appear at the border at the image, and keeping it in such a position does not seem to be very clever, while with the first control law it easily can be brought to the image center.

## 4 Application to fixation in an alignment task

The aim of our second task is to position the image plane parallel to an observed plane while ensuring a fixation task, such that  $P_c = P_i$  (see Fig 10). Such a fixation is important in order that the object always appear in the image despite the rotational motion involved by the alignment.

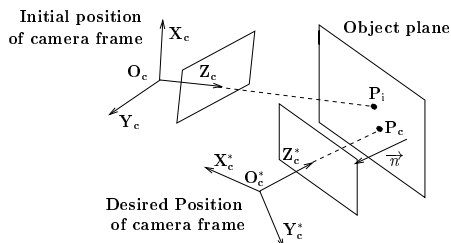


Figure 10: Task to be performed

### 4.1 Control laws

It can be denoted from (2) that the quadratic terms of the motion model show the terms  $\gamma_1$  and  $\gamma_2$  expressing the angular position of the observed plane relatively

to the camera [8]. Thus, it appears that, with a non-null motion along the optical axis, the alignment task will be solved if and only if the following condition is respected:

$$\begin{pmatrix} b_1 + \Omega_y \\ b_2 - \Omega_x \end{pmatrix} = - \begin{pmatrix} \gamma_1 v_z \\ \gamma_2 v_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Several approaches can be considered to solve the problem of adding a fixation task to the alignment one. In all the cases considered below, translation  $T_z$  along the optical axis is considered constant and rotation  $\Omega_z$  around this axis is considered null, as it has no influence upon either the alignment nor the fixation. Furthermore, all control laws are established considering an exponential decay of the error.

The first approach is to consider that the image center will always correspond to the same point from the object if its apparent speed is zero, what means  $(a_1, a_4) = (0, 0)$  (see (2)). Then, a motion-based visual control law can be established considering the task constraining the following vector  $s$  to reach a zero value:

$$s = ((a_1, a_4, b_1 + \Omega_y, b_2 - \Omega_x)^T$$

Neglecting the second and upper order terms, the interaction relation linking the derivative  $\dot{s}$  of  $s$  with the motion of the camera can be expressed under the following form [20]:

$$\dot{s} = \begin{bmatrix} -1/Z_p & 0 & 0 & -v_z \\ 0 & -1/Z_p & v_z & 0 \\ 0 & 0 & 0 & -v_z \\ 0 & 0 & v_z & 0 \end{bmatrix} \begin{pmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{\Omega}_x \\ \dot{\Omega}_y \end{pmatrix} + v_z \begin{pmatrix} -s_1 \\ -s_2 \\ s_3 \\ s_4 \end{pmatrix} \quad (3)$$

This leads to the the following rotational velocity and translational acceleration control law:

$$\begin{pmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{\Omega}_x \\ \dot{\Omega}_y \end{pmatrix} = - \begin{bmatrix} -T_z & 0 & T_z & 0 \\ 0 & -T_z & 0 & T_z \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{pmatrix} \frac{\lambda - v_z}{v_z} s_1 \\ s_2 \\ \frac{\lambda + v_z}{v_z} s_3 \\ s_4 \end{pmatrix} \quad (4)$$

A second approach, previously presented in [8], is to consider separately the alignment and the fixation tasks. Once the alignment control law is designed, the fixation can be obtained by direct compensation of the rotational motion using a translational one. It may be done by an open loop control in order to maintain  $a_1$  and  $a_4$  to zero using:

$$\begin{pmatrix} a_1 \\ a_4 \end{pmatrix} = 0 \Leftrightarrow \begin{cases} v_x = -\Omega_y \\ v_y = \Omega_x \end{cases}$$

Since  $\Omega_x$  and  $\Omega_y$  are controlled by the alignment task,  $T_x$  and  $T_y$  can be used for fixation and compensation

of the rotational motion. However, this approach leads to an open control loop which does not take into account a possible error in the estimation of  $Z_p$  (needed to deduce  $T_x$  and  $T_y$  from  $v_x$  and  $v_y$ ), nor than drift due to non zero affine and quadratic terms when the initial center does not appear at the image center anymore.

A third and new approach is to retrieve displacement  $(x, y)^T$  due to rotational motion by integrating speed in the image, and to regulate it to zero by translational motion.

The vector of measures  $s$  for the alignment and fixation task can thus be chosen as:

$$s = (x, y, b_1 + \Omega_y, b_2 - \Omega_x)^T$$

In this case, the interaction relation between  $\dot{s}$  and the camera motion is given by the following equation [8, 20] where the same approximation to the first order has been considered:

$$\dot{s} = \begin{bmatrix} -1/Z_p & 0 & 0 & -1 \\ 0 & -1/Z_p & 1 & 0 \\ 0 & 0 & 0 & -v_z \\ 0 & 0 & v_z & 0 \end{bmatrix} \begin{pmatrix} T_x \\ T_y \\ \Omega_x \\ \Omega_y \end{pmatrix} + v_z s$$

The corresponding control law is thus given by:

$$\begin{pmatrix} T_x \\ T_y \\ \Omega_x \\ \Omega_y \end{pmatrix} = \frac{\lambda + v_z}{v_z} \begin{bmatrix} -T_z & 0 & Z_p & 0 \\ 0 & -T_z & 0 & Z_p \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} s$$

This control law is similar to (4) (when replacing  $\dot{T}$  by  $T$  and  $(a_1, a_4)$  with  $(x, y)$ ). It can be easily explained noticing that the model of motion considered is  $(\dot{x}, \dot{y})^T = (a_1, a_4)^T + v_z(x, y)^T$  and that  $\dot{v}_z = -v_z^2$  [20]. Thus, relation (3) can also be deduced by derivating the previous one.

## 4.2 Results

These control laws have been tested on our experimental robotic cell. The initial image of the observed plane is presented in Fig 11. Experiments have been made with the same initial position for every case, where angular errors are about 30 degrees on each axes. As the estimation of the quadratic parameters of motion is quite costly, rate of control is about 1 Hz and the decreasing parameter  $\lambda$  was equal to 0.04. Results obtained for the alignment are displayed in [8]. We only present here results for the fixation.

Concerning the first control law, it appeared that trying to control by acceleration with such a rate is

nearly impossible. Speed reaches a too great value before it can be updated by a new acceleration. Thus, we present results only for the second and third control laws. For the second one, the displacement of the image center along time has been estimated to be compared with the one obtained in the third case, but has not been taken into account in the control law. In each case, the number of iteration taken into account is the necessary one so that angular errors reach a continuous running at less that one degree.

Here, we present for each control law, the estimated drift of the initial center (see Fig. 12, 14) and  $a_1$  and  $a_4$  (see Fig. 13, 15).



Figure 11: Alignment and fixation task: initial image

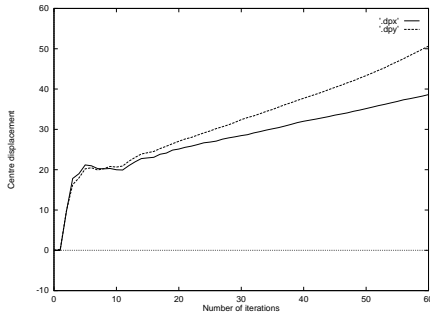


Figure 12: Open loop: Drift of initial centre (in pixel)

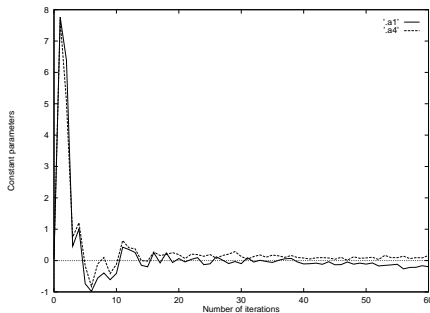


Figure 13: Open loop: Constant parameters

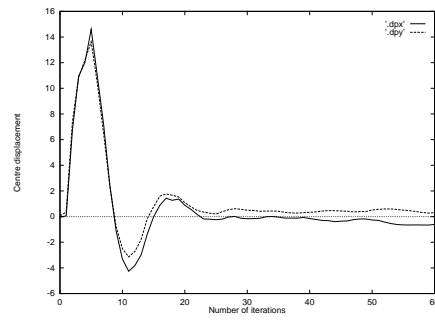


Figure 14: Closed loop: Drift of initial center (in pixel)

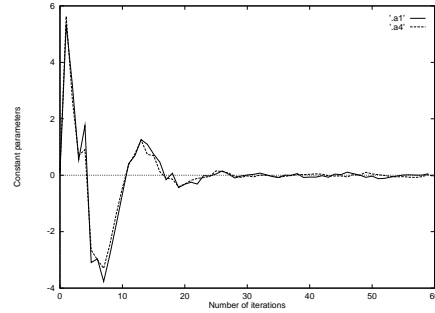


Figure 15: Closed loop: Constant parameters

The point projected at the center of the initial image has been “manually” retrieved on the final images for the two cases. In the open loop scheme, respective drifts on  $x$  and  $y$  axes are about 35 and 48 pixels whereas, in the closed loop one, they are only about 1 and 3 pixels. These results, similar to the estimated positions (which again validates the accuracy of the estimation scheme), proves that the second control law answers better to the desired behavior. This is due to the fact that the first control law only ensures the constant parameters to reach a zero value (which is done after the 15<sup>th</sup> iterations), but does not compensate drift of the center obtained during these first iterations. Then, after iteration 15, this center is subjected to the divergent motion brought by the translation along the optical axis. On the contrary, as the second control law is directly based on this estimated position, it is obviously brought to zero. There, we can notice that the constant parameters are also null at convergence, but drift appearing in the 5 first iterations is compensated in the following 5 ones.

## 5 Conclusion

The aim of this paper was to prove that image-based control can be done by integrating dynamic features. Usually, when measurements are integrated along time to estimate a value knowing its successive derivatives, a problem of bias in the estimation appears, due to the inevitable noises. Here, we showed that this problem

does not appear. Tasks such as tracking a moving object or gazing at the same point of the scene when the camera is in motion, can be fulfilled using this technique. This is done by retrieving position from speed in the image and then applying control laws developed for image-based servoing.

## References

- [1] 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. on Robotics & Automation*, 9(2):152–165, Apr. 1993.
- [2] 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 H. Wjshof, editor, *Computer Science in the Netherlands*, pages 52–63. CWI, Amsterdam, Nov. 1993.
- [3] 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*, volume 1, pages 181–187, Pittsburgh, Aug. 1995.
- [4] F. Chaumette and A. Santos. Tracking a moving object by visual servoing. In *12<sup>th</sup> World Congress IFAC*, volume 9, pages 409–414, Sydney, Australia, July 1993.
- [5] C. Colombo, E. Kruse, A.M. Sabatini, and P. Dario. Vision-based relative positioning through active fixation and contour tracking. In *Int. Symposium on Intelligent Robotic Systems*, Grenoble, July 1994.
- [6] P.L. Corke and M.C. Good. Controller design for high performance visual servoing. In *12th World congress IFAC*, volume 9, pages 395–398, Sydney, Australia, July 1993.
- [7] E. Coste-Manière, P. Couvignon, and P.K. Khosla. Visual servoing in the task-function framework: a contour following task. *Journal of Intelligent Robotic Systems*, 12(1):1–22, Jan. 1995.
- [8] 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*, volume 1, pages 43–48, Grenoble, France, Sept. 1997.
- [9] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics & Automation*, 8(3):313–326, June 1992.
- [10] G. Hager. The x-vision system: a general purpose substrate for real-time vision-based robotics. In *IEEE Workshop on Vision for Robots*, pages 56–63, Pittsburgh, USA, Aug. 1995.
- [11] K. Hashimoto, editor. *Visual servoing. Real-time control of robot manipulators based on visual sensory feedback*. World scientific series in robotics and automated systems. World scientific, 1993.
- [12] K. Hashimoto, T. Ebine, K. Sakamoto, and H. Kimura. Full 3D visual tracking with nonlinear model-based control. In *American Control Conference*, pages 3180–3185, San Francisco, California, June 1993.
- [13] S. Hutchinson, G. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics & Automation*, 12(5):651–670, Oct. 1996.
- [14] P. Martinet, F. Berry, and J. Gallice. Use of first derivative of geometric features in visual servoing. In *IEEE Int. Conf. on Robotics & Automation*, volume 4, pages 3413–3419, Minneapolis, MN, Apr. 1996.
- [15] P. Nordlund and T. Uhlin. Closing the loop: detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, 14(4):265–275, May 1996.
- [16] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, Dec. 1995.
- [17] N.P. Papanikolopoulos, B. Nelson, and P.K. Khosla. Six degree-of-freedom hand/eye visual tracking with uncertain parameters. *IEEE Trans. on Robotics & Automation*, 11(5):725–732, Oct. 1995.
- [18] P. Questa, E. Grossmann, and G. Sandini. Camera self orientation and docking maneuver using normal flow. In *SPIE AeroSense'95*, Orlando, Florida, Apr. 1995.
- [19] C. Samson, M. Le Borgne, and B. Espiau. *Robot control : the task function approach*. Oxford University Press, 1990.
- [20] M. Subarrao and A. Waxman. Closed-form solutions to image equations for planar surface in motion. *Computer Vision, Graphics, and Image Processings*, 36(2):208–228, Nov. 1986.
- [21] V. Sundareswaran, P. Bouthemy, and F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *International Journal of Robotics Research*, 15(6):629–645, Dec. 1996.