Image-based visual servoing on planar objects of unknown shape

Christophe Collewet, François Chaumette, P. Loisel

HAL Id: inria-00352119

https://hal.inria.fr/inria-00352119

Submitted on 12 Jan 2009

Image-based visual servoing on planar objects of unknown shape

Christophe Collewet

François Chaumette

Philippe Loisel

Cemagref
17 Avenue de Cucillé
35044 Rennes Cedex, France
christophe.collewet@cemagref.fr

IRISA / INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
francois.chaumette@irisa.fr

Cemagref
17 Avenue de Cucillé
35044 Rennes Cedex, France
philippe.loisel@cemagref.fr

Abstract

This paper proposes a way to achieve positioning tasks by visual servoing, for any orientation of the camera, when the desired image of the observed object cannot be precisely described. The object is assumed to be planar and motionless but no knowledge about its shape or pose is required. To simplify the problem, first, we treat the case of a threadlike object and then we show how our approach can be generalized to an object with three particular points. The control law is based on the use of 2d visual servoing and on an estimation of a 3d parameter. Experimental results relative to objects of unknown shape are given to validate the approach. In addition, an algorithm to estimate the depth between the object and the camera is provided which leads to the dimensions of the object.

1 Introduction

Visual servoing is now a classical technique in robot control (see [1] for a description of the different approaches). Nevertheless, in the most often case of an "eye-in-hand" system [2, 3, 4], we still cannot achieve positioning tasks with regard to deformable or not well known objects. Such a case appears when we have to treat applications for example in surgical domain, agri-food industry, agriculture or in unknown environments (underwater, space). Indeed, except manufactured goods for which a model often exists, we rarely have a precise description of the object or of the desired visual features.

In the case of a 3d visual servoing two approaches exist. The first one and the most often used, is based on the computation of the pose object/camera and then requires a model of the object. Therefore, this approach cannot be used in our case. The second one is based on 3d reconstruction by dynamic vision [5, 6] but these techniques are currently not accurate enough

with regard to the errors of reconstruction. On the other hand, active vision [7, 8, 9, 10, 11] can limit such errors. However, this approach has only been used on very simple objects. Moreover, let us point out that those 3d reconstruction techniques are sensitive to the calibration of the system.

On the other hand, lots of work in 2d visual servoing have shown that the closed-loop system is little sensitive to calibration problems [4, 12, 13]. Similar results have been obtained concerning the recent 2 1/2d approach [14]. However, the 2d approach as well as the 2 1/2d cannot cope with the objects being studied. Indeed, let us consider an accurate positioning task related to such objects, a raw ham for example. Even though these objects are part of a same class, they are different enough not to use unique desired visual features for all the objects in the class. In such applications the desired visual features have to be considered as unknown. Few authors relate such cases. In [15], thanks to dynamic visual features a positioning task consisting in moving the camera in front of a planar object of unknown shape can be achieved. However, such an approach needs particular motion parameters estimation leading currently to high computation duration and, consequently, to a low control scheme rate. Moreover, this approach does not well suit for positioning tasks since a motion is necessary.

The method described in our paper is based on the use of points as visual features, it can achieve positioning tasks when any orientation is required and for the same objects as those used in [15]: planar objects of unknown shape. Our approach needs no assumption about the shape of the observed object but we have to assume the object motionless. The approach is based on the use of 2d visual servoing and on an estimation of a 3d parameter that will be precised below. We will see that this structure maintains the object of interest in the field of view of the camera, does not need a calibrated camera although it provides dimensions of the object and an estimation of the depth. Unfortunately,

247

it is sensitive to the calibration of the robot. Finally, our method combines in part the advantages of both 2d visual servoing and 3d reconstruction.

This paper is organized as follows: in Section 2, we deal with a threadlike object to achieve the particular positioning task consisting in moving the robot in front of this object. Next, we show in Section 3 how this approach can be readily generalized to an object with three particular points and for any orientation of the camera. Finally, experimental results are given to validate the algorithm.

2 Case of a simple object: the segment

2.1 Task specification

In this section, we propose to move a robot in front of a threadlike object by visual servoing. In order to achieve applications as those described in the previous section, we suppose the length of the object unknown as well as its pose. The object is described by a segment $[m_1, m_2]$ with $\underline{Om_1} = (x_1, y_1, z_1)^T$ and $\underline{Om_2} = (x_2, y_2, z_2)^T$, expressed in the camera frame which is centered in the optical center O. The optical axis of the camera is the z axis of the camera frame (see Figure 1). This object projects on the image plane by a perspective projection as a segment $[M_1, M_2]$ with $\underline{OM_1} = (X_1, Y_1, f)^T$ and $\underline{OM_2} = (X_2, Y_2, f)^T$ according to:

$$\underline{OM} = \frac{f}{z} \underline{Om} \qquad (1)$$

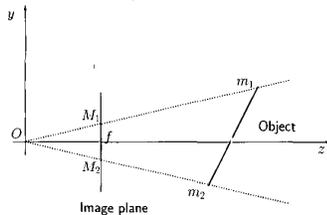where f is the focal length assumed to be equal to 1.

Figure 1: Projection of the object on the image plane

The task is achieved, *i.e.* the camera is in front of the object, when the optical axis is orthogonal to $[m_1, m_2]$ and the image centered with respect to the y axis. In this case, we have $z_1 = z_2 = z^*$ and $Y_1 = -Y_2$.

Even though the object is very simple, the task is not obvious to achieve since an infinity of object poses leads to a centered image without achieving the task. Nevertheless, a way to achieve it is to center the image

and then, to scan other positions. This is a typical case where the *task function approach* described in [2] has to be used. So, we will consider as the *main task* the one which maintains centered the image, and the *secondary task* the one which scans other positions while ensuring the image centered.

2.2 Control law

2.2.1 Main task

A way to center the image is to choose as visual features the following vector:

$$\underline{s} = (X_1, X_2, Y_1 + Y_2)^T \qquad (2)$$

and as the desired visual features the vector:

$$\underline{s}^* = (0, 0, 0)^T \qquad (3)$$

We immediately obtain the interaction matrix at the desired position [1, 2]:

$$L_{\underline{s}^*}^T = \begin{pmatrix} -\dfrac{1}{\widehat{z^*}} & 0 & 0 & 0 & -1 & Y_1 \\ -\dfrac{1}{\widehat{z^*}} & 0 & 0 & 0 & -1 & -Y_1 \\ 0 & -\dfrac{2}{\widehat{z^*}} & 0 & 2\left(1 + Y_1{}^2\right) & 0 & 0 \end{pmatrix} \qquad (4)$$

where $\widehat{z^*}$ is an approximation of z^*.

2.2.2 Secondary task

At this step, we want to equate z_1 to z_2 while ensuring the image centered. Therefore, such a motion has to belong to Ker $L_{\underline{s}^*}^T$ to ensure $\underline{s} = \underline{s}^*$. Thereafter, we can apply a control law of the following form to achieve the task:

$$T_c = -\lambda L_{\underline{s}^*}^{T^+} \left(\underline{s} - \underline{s}^*\right) + T_s \qquad (5)$$

with:

$$T_s = \begin{pmatrix} 0 & \widehat{z^*}\omega_x \left(1 + Y_1{}^2\right) & 0 & \omega_x & 0 & 0 \end{pmatrix}^T \qquad (6)$$

which belongs to Ker $L_{\underline{s}^*}^T$ and in which ω_x has to be determined.

If $\underline{s}$ is in the neighborhood of $\underline{s}^*$, only T_s generates a motion. Intuitively, it is easy to see that the length of the segment $l = Y_1 - Y_2$ may supply useful information. In particular, we think it takes a maximum when the camera is in front of the object. In order to verify this assumption and to obtain all the parameters needed for (5), we want to know how l changes during the motion of the camera. Besides, we think that the trajectory of the camera can supply interesting information.

2.3 Modeling

2.3.1 Modeling of the length of the segment

Let us consider the points $m_1 = (y^*, z^*)^T$, $m_2 = (-y^*, z^*)^T$ and $m_0 = (0, z_0)^T$ expressed in the camera frame when the camera is at the desired position. We consider that the motion during the maximization of l results from a x axis rotation of angle θ_x, centered in m_0 (see Figure 2). Therefore, m_1 and m_2 can be expressed as: $m_1\prime = \mathcal{R}(m_1 - m_0) + \mathcal{T}$ and $m_2\prime = \mathcal{R}(m_2 - m_0) + \mathcal{T}$ which yields the projection $M_1 = (X_1, Y_1)^T$ and $M_2 = (X_2, Y_2)^T$ of $m_1\prime$ and $m_2\prime$ respectively. Then, we search the solution of $Y_1 = -Y_2$ with respect to z_0 to perform a motion which belongs to Ker $L^T\big|_{\underline{s}=\underline{s}^*}$, as seen in 2.2.2. This expression is complicated. Nevertheless, when $v = y^*/z^*$ is low, it can be expressed at 2nd order by:

$$z_0 = z^* \tag{7}$$

while l becomes simply at 3rd order:

$$l_m = 2v \cos \theta_x \tag{8}$$

2.3.2 Modeling of the trajectory of the camera

Thanks to (7), we can derive the trajectory of the camera with respect to the desired camera position (see figure 2):

$$\begin{cases} z &= z^*(1 - \cos \theta_x) \\ y &= z^* \sin \theta_x \end{cases} \tag{9}$$

Thus the trajectory is simply a circle centered in $(0, z^*)$ with a radius z^*:

$$y^2 - (z - z^*)^2 = z^{*2} \tag{10}$$

2.3.3 Application to control

During a motion, the form of which is given by the control law (5), we proceed to an on-line learning of l, supposed to be modelized by (8). In practice, the function we need is $l_m(\Theta_x - \Theta_{xc})$ with $\Theta_x - \Theta_{xc} = \theta_x$ (remember that (8) is obtained with respect to the desired position) and thereafter, Θ_{xc} is the unknown value where l_m takes its maximum. $l_m(\Theta_x - \Theta_{xc})$ yields a linear expression in $\cos \Theta_x$ and $\sin \Theta_x$ that provides Θ_{xc} thanks to a least squares algorithm. Moreover, to ensure an exponential decay of the angular error $\Theta_x - \Theta_{xc}$, we impose:

$$\omega_x = K_x(\Theta_x - \Theta_{xc}) \tag{11}$$

with K_x a positive value.

At the begining of the motion, when the estimation supplies a not accurate enough value of Θ_{xc}, we fix ω_x to a constant value ω_0.
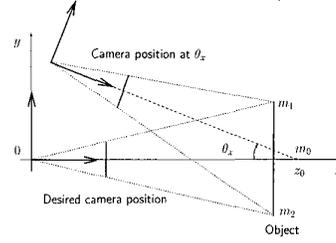

Figure 2: Modeling of the length of the segment with respect to θ_x

remark: Note that Θ_x is obtained thanks to the odometry of the robot and therefore does not depend of the calibration of the camera. Of course, it depends on the calibration of the robot.

On the other hand, it is thus possible to extract $\widehat{z^*}$ from the trajectory of the robot thanks again to an on-line nonlinear algorithm. It must lead to low errors $\|\underline{s} - \underline{s}^*\|$ during the motion defined by (5). Moreover, thanks to the perspective transformation, 3d features can be obtained as will be shown in Section 3.2.

After having treated the case of segments, we will show, in the next section, how our method can be generalized to a planar object of unknown and complex shape when at least three feature points can be extracted from the image.

3 Case of a planar object with at least three feature points

3.1 Specifying the main task

In section 2, the task has been achieved by maximization of the length of the object image. In the case of a 2d object, a natural transposition consists in maximizing in the image its area S. However, obtaining its analytical expression in the general case seems to be out of reach. Thus, we simplify the approach by assuming that three feature points exist and then the image becomes simply a triangle with vertices $M_1(X_1, Y_1)$, $M_2(X_2, Y_2)$ and $M_3(X_3, Y_3)$. In the same way we have modelized l in 2.3.1, we have tried to express a modeling of S. Unfortunately, this study failed. Nevertheless, thanks to the results of section 2, we can maximize sequentially l then $h = X_3 - X_1$. Thus, we impose $Y_1 = -Y_2$ with regard to the maximization of l and $X_1 = -X_3$ with regard to the maximization of h. If we want to ensure that vectors of the kernel of the interaction matrix related to $\underline{s} = (X_1 + X_3, Y_1 + Y_2)^T$ are of the form of (6), we have to impose $Y_1 = Y_3$ and $X_1 = X_2$. Unfortunately these two constraints cannot

be satisfied simultaneously in the case of any triangle. Therefore, we have arbitrarily chosen to impose $X_1 = X_2$ yielding:

$$\underline{s} = (X_1 - X_2, X_1 + X_3, Y_1 + Y_2)^T \qquad (12)$$

$$\underline{s}^* = (0, 0, 0)^T \qquad (13)$$

$$L_{\underline{s}^*}^T = \begin{pmatrix} 0 & 0 & 0 & 2X_1Y_1 & 0 & 2Y_1 \\ -2/\widehat{z^*} & 0 & 0 & X_1(Y_1 - Y_3) & -2\left(1 + X_1^2\right) & Y_1 + Y_3 \\ 0 & -2/\widehat{z^*} & 0 & 2\left(1 + Y_1^2\right) & 0 & -2X_1 \end{pmatrix} \qquad (14)$$

and the two vectors which belong to Ker $L_{\underline{s}^*}^T$:

$$\underline{T}_{sl} = (-\widehat{z^*}\omega_x X_1 Y_3, \ \widehat{z^*}\omega_x(1 + X_1^2 + Y_1^2), \ 0,$$
$$\omega_x, \ 0, \ -\omega_x X_1)^T \qquad (15)$$

$$\underline{T}_{sh} = \begin{pmatrix} -\widehat{z^*}\omega_y\left(1 + X_1^2\right) & 0 & 0 & 0 & \omega_y & 0 \end{pmatrix}^T \qquad (16)$$

We will see that even if (15) is different from (6), satisfactory results are obtained. The control law is thus:

$$T_c = -\lambda L_{\underline{s}^*}^{T^+}(\underline{s} - \underline{s}^*) + T_{sl} + T_{sh} \qquad (17)$$

In the next section we present the behavior of this control law first on a simple object and next on a more complicated object.

3.2 Experimental results

3.2.1 Experiments on a simple object

The first experiment consists in moving the camera in front of an object with 3 feature points. To simplify the image processing, a binary object has been used. Figures 3a (4a), 3b (4b), 3c (4c) and 3d (4d) depict respectively the behavior of the components of T_c, the error in the sensor $\|\underline{s} - \underline{s}^*\|$, the desired and current angular values and the measured and modelized image length during the maximization of l (h). Figure 5 summarizes the different phases to achieve the task. Figure 3d (4d) confirms the theorical results about the modeling of $l = f(\theta_x)$ ($h = f(\theta_y)$). Moreover, on Figure 3c (4c) we remark that we quickly obtain the desired angular value which refines during the estimation and finally stabilizes. In addition, thanks to the least squares algorithm, lets us point out that the noise has not a lot of effect on this value. In other respects, we have applied DeMenthon's method [16] to obtain the poses of the camera before and after servoing with respect to the object (4 points are then required). The initial pose was around $(25^o, 19^o, 11^o)$ according to the x, y and z axes of the camera. The positioning error was around 0.5^o. Therefore, these results are satisfactory.

Besides, as seen in 2.3.2 we have access to $\widehat{z^*}$. This algorithm yielded a value of 706.1 mm while DeMenthon's method gave in desired position $z_1 = 707.9$ mm, $z_2 = 708.6$ mm and $z_3 = 709.9$ mm. The consequence of a good value for $\widehat{z^*}$ can be seen on Figure 4b since the error decreases suddenly. Thereafter, once the task is achieved, the camera is in front of the object and $\widehat{z^*}$ is known. Then, according to (1) 3d features can be reconstructed. For example for l and h the following values have been obtained: $\widehat{l} = 13.06$ cm and $\widehat{h} = 13.05$ cm instead of $l = h = 13$ cm.

To validate those first results, 20 other experiments has been led for different initial positions (into [-4°, 22°] x [8°, 33°] x [-25°, 36°] with respect to the rotations of axes x, y and z). Very good results have been obtained concerning as well the positioning task as the reconstruction of 3d features. They are summarized in the table 1 in which the function $Q(x)$ means the absolute value of the relative error on the measure of x in percent.

Table 1: Results on 20 experiments.

x	$\overline{m} = 1/N \sum_i x_i$	max $|x_i|$	σ_x	min $|x_i|$
γ	-0.009	0.439	0.007	0.017
β	-0.489	0.751	0.156	0.133
$Q(l)$	0.576	1.103	0.136	0.246
$Q(h)$	0.562	1.104	0.140	0.242
$Q(z)$	0.562	1.109	0.145	0.240

In addition, errors have been introduced in the intrinsic parameters of the camera: we have added an error of 10 % on these parameters and we have neglected the radial distorsion of the lens. Experimental results, consisting in achieving positioning tasks, showed that non significant difference exists concerning the positioning error between a calibrated camera and a coarse calibrated one. This result agrees with the remark made in section 2.3.3.

The second experiment consists in positioning the camera not in front of the object but for a particular desired pose. Since we know the angular value Θ_{xc} (Θ_{yc}) (see 2.3.3) which leads to move the robot in front of the object, by adding specified constant values any orientations can be reached (see Figures 6 and 7). In this case we obtained (-15.2^o, -19.0^o) instead of (-15°, -20°). Again, these results are satisfactory.

3.2.2 Experiment on a complicated object

In this third experiment, we consider the case of a raw ham. This object is moving along a conveyor and the goal is to place the camera in front of it to perform

an inspection of its cut by a machine vision. To do this, we used the method described in [17] to select and track the 3 feature points we need. It is based on SSD matching and assumes translational frame-to-frame displacements. The experimental results are depicted on Figures 8 and 9. They lead to similar conclusions as the previous experimental results. However, the algorithm of extraction of the visual features is more noisy and a higher positioning error is obtained but, this time, difficult to measure. Again, 3d features can be obtained to characterize the cut of the ham.

4   Conclusion

We have presented in this paper a way to achieve any 2d visual servoing positioning task in the case of a planar and motionless object of complex shape thanks to simple geometric visual features. Experimental results have shown that accurate positionings can be obtained ($\approx 0.5^{o}$) even if the camera were not calibrated. In addition, thanks to the trajectory of the camera, the depth between the camera and the object can be obtained yielding precise 3d reconstruction (≈ 0.6 %). Thus, our method combines in part the advantages of both 2d and 3d visual servoing in the sense that 3d information can be obtained with a coarse calibrated camera.

Finally, we think that this work contributes to an expansion of the application area of visual servoings in the sense that now, complex objects or objects of unknown shape can be treated even if the desired image is not precisely known.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[2] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.

[3] N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Vision and control techniques for robotic visual tracking," in *IEEE Int. Conf. on Robotics and Automation, ICRA'91*, Sacramento, USA, April 1991, pp. 857–864.

[4] H. Hashimoto, T. T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *IEEE Int. Conf. on Robotics and Automation, ICRA'91*, Sacremento, California, USA, April 1991, pp. 2267–2272.

[5] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images, a review," *Proc. of IEEE*, vol. 76, pp. 917–935, August 1988.

[6] G. Adiv, "Inherent ambiguities in recovering 3d motion and structure from a noisy flow field," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 477–489, May 1989.

[7] Y. Aloimonos, I. Weiss, and A. Bandopadhay, "Active vision," *Int. Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, January 1987.

[8] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 996–1005, August 1988.

[9] G. Sandini and M. Tistarelli, "Active tracking strategy for monocular depth inference over multiple frames," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 13–27, January 1990.

[10] K. Kutulakos and C. Dyer, "Recovering shape by purposive viewpoint adjustment," *Int. Journal of Computer Vision*, vol. 12, no. 2, pp. 113–136, February 1994.

[11] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 492–504, May 1996.

[12] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Third Int. Symp. on Experimental Robotics, ISER'93*, Kyoto, Japan, October 1993.

[13] G. D. Hager, W. C. Chang, and A. S. Morse, "Robot hand-eye coordination based on stereo vision," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 30–39, February 1995.

[14] E. Malis, F. Chaumette, and S. Boudet, "2 1/2d visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[15] A. Crétual and F. Chaumette, "Positioning a camera parallel to a plane using dynamic visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'97*, Grenoble, Septembre 1997, pp. 43–48.

[16] D.F. DeMenthon and L.S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995.

[17] J. Shi and C. Tomasi, "Good features to track," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, USA, June 1994, pp. 593–600.
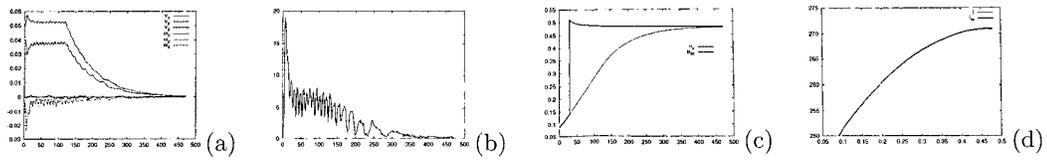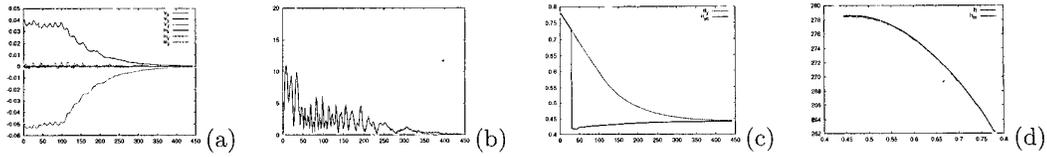
Figure 3: 1st experiment. Maximization of l.

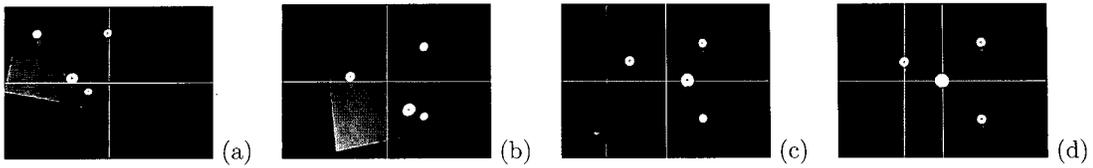Figure 4: 1st experiment. Maximization of h.

Figure 5: 1st experiment. (a) Initial Position. (b) Position after "centering". (c) Position after maximization of l. (d) Final position.
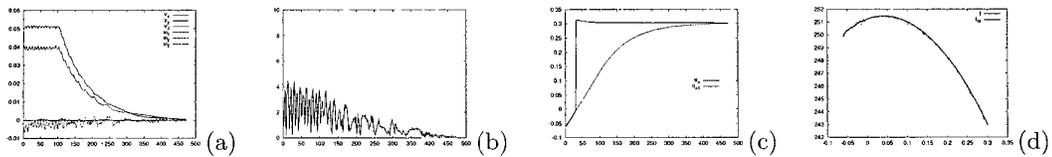

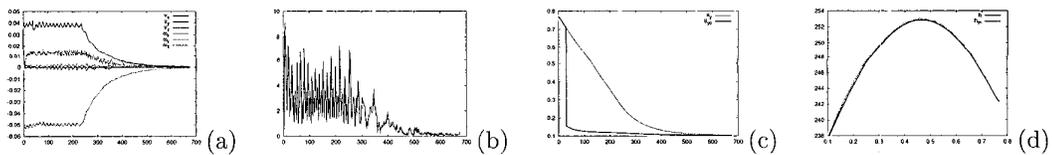Figure 6: 2nd experiment. Servoing of Θ_x.

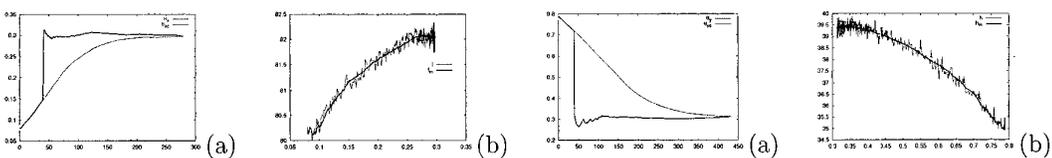Figure 7: 2nd experiment. Servoing of Θ_y.

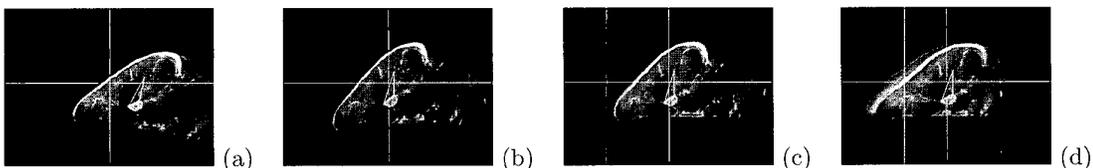Figure 8: 3rd experiment. Maximization of l and h.

Figure 9: 3rd experiment. (a) Initial Position. (b) Position after "centering". (c) Position after maximization of l. (d) Final position.

252