B-1d-09-6

POSE-ESTIMATION AND CONTROL IN A 3D VISUAL SERVOING SYSTEM

M. Vargas, F.R. Rubio and A.R. Malpesa[1]

Dpto. Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros, Universidad de Sevilla
Camino de los Descubrimientos s/n, 41092-Sevilla (Spain)
Ph.:+34954487350, Fax:+34954487340, E-mail:rubio@esi.us.es

Abstract: The present work studies the case of a 3D visual servoing system composed by a robot manipulator, a CCD camera rigidly mounted on the end-effector of the robot, and other computer-based hardware. The aim of the system is to keep a desired relative position and orientation between the object and the camera, compensating for the autonomous object motion. Two existing algorithms for object's pose estimation have been implemented. Several controllers are designed and its performance evaluated along the paper. *Copyright © 1999 IFAC*

Keywords: Robotic manipulators, Matrix equations, Image processing, Control system design, Digital Signal Processor

1. INTRODUCTION

The interaction between vision sensors and robotic systems has been subject of great effort in the recent research. It has special interest in applications where a robot has to interact with moving parts or objects, whose position and orientation is unpredictable or not known with precision. The term *visual servoing* is applied to visually guided systems that make use of one or several, static or moving cameras to provide visual information as a feedback signal to control the manipulator in accomplishing its task. Some examples of applications for visual servoing are: fruit picking, teleoperation, missile tracking, pursuer autonomous vehicles, etc.

The present work studies the case of a single-camera, visual servoing system using the *hand-eye configuration*, composed by a robot manipulator, a CCD camera rigidly mounted on the end-effector of the robot, and other hardware components that will be detailed later in the paper. The aim of the system is to keep a desired relative position and orientation between the object and the camera, compensating for the autonomous object motion. The implementation and performance analysis of this system in the planar case can be found in a previous work (Vargas *et al.*, 1998). In the present paper, the extension to the 6-DOF case is presented.

Corke and other authors (Corke, 1996), (Hashimoto, 1993), distinguish several approaches when handling a visual servoing implementation. The one used here is the *Position-Based Approach*, in which visual features are used to estimate the relative object-camera position, that is actually used to control the system.

Since one of the critical issues in the position-based approach is the estimation of the object position and orientation relative to the camera, two different existing methods have been essayed in this work: the first one is a variation of the Tsai's method (Tsai, 1987), by Sørensen (Sorensen, 1994), and the other one is an in-

[1] Partially supported by CICYT, under grant TAP 98-0541

 ISBN: 0 08 043248 4

cremental method due to Heuring and Murray (Heuring and Murray, 1996*b*), (Heuring and Murray, 1996*a*).

The structure of the paper is the following: First of all, in section 2, the deduced model of the complete 6-DOF visual servoing system is detailed. In section 3, the two different, existing methods, used for the relative object-camera pose estimation, are mentioned. After that, in section 4, the design of several controllers is handled, starting from a simplified model of the system. This section includes real experiments to compare the response of each controller and the performance using both pose estimation methods. Finally, the paper describes the specific components of our system in section 5, and concludes with a summary in section 6.

2. VISUAL SERVOING MODEL

One of the main problems in the case of the position-based approach, is the precise estimation of the relative object-camera pose (position and orientation, entailing up to 6 coordinates, which can be handled in the form of homogeneous matrices). The model used in the present work was built up starting from the one-dimensional case as described in detail in (Vargas *et al.*, 1998).

Briefly, the system can be viewed at two levels:

• **Low-level Model:** That models the robot arm behaviour. The robot manipulator used is the *PUMA 560*, commanded via serial port, in cartesian coordinates, making use of the *ALTER* facility (Unimation, 1986). The model derived in (Vargas *et al.*, 1998), for each cartesian degree of freedom, is shown in figure 1. Where, the sampling time is $T_1 = 28ms$ and:

$$G_{arm}(s) = \frac{1}{(1+\tau_1 s)(1+\tau_2 s)} \qquad (1)$$

With: $\tau_1 = 0.003$ and $\tau_2 = 0.005$

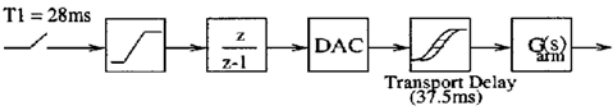

Fig. 1. Detailed robot model

• **High-level Model:** Several coordinate systems and geometric transformations are involved in this model. In figure 2, the position of the robot arm, camera and object are shown in two consecutive sampling intervals.

In this figure, $\{W\}$ is the world coordinate frame, usually at the base of the robot. $\{O\}$ is the coordinate frame bound to the moving object (time-varying $\{O_k\}$, k: sampling period). $\{C_k\}$ is the camera coordinate system, with axes X and Y parallel to the horizontal and vertical image plane

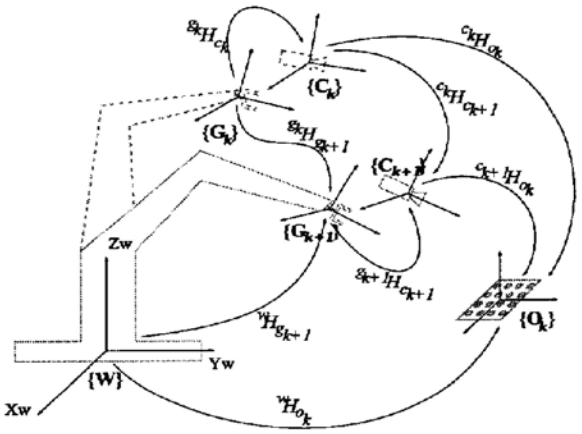

Fig. 2. Nomenclature of the coordinate systems and matrix transformations

axes, and the Z axis, passing through the center of the camera lens. $\{G_k\}$ is the gripper (or robot end-effector) coordinate system.

The high-level model describes the behavior of the full 6-DOF version of our servoing system, and can be seen in figure 3.

In this block diagram, the object position is an external, unpredictable input: ${}^{W}H_{O_{k+1}}$ [2]. This is compared with the current camera pose, ${}^{W}H_{C_{k+1}}$, giving the relative object-camera pose:

$${}^{C_{k+1}}H_{O_{k+1}} = \left[{}^{W}H_{C_{k+1}}\right]^{-1} \cdot {}^{W}H_{O_{k+1}} \qquad (2)$$

The image capturing mechanism is modeled as the *World-To-Screen* module in the figure, and converts the set of feature points present on the real object to their corresponding positions in the frame-grabber memory. Then, after some image processing and algebraic computations, included in the block *Pose-Estimation* (which the following section is devoted to), the estimated object-camera pose is obtained: ${}^{C_{k+1}}\hat{H}_{O_{k+1}}$. This estimated pose is compared with the desired relative pose, giving the necessary camera displacement and rotation:

$$\Delta H_{C_k} = {}^{C_k}H_{C_{k+1}} = {}^{C_k}\hat{H}_{O_k} \cdot {}^{O_k}H_{C_{k+1}} = {}^{C_k}\hat{H}_{O_k} \cdot H_d{}^{-1} \qquad (3)$$

Then, the camera-gripper transformation (${}^{G}\hat{H}_{C}$, which has been estimated off-line, using some calibration method (Tsai and Lenz, 1989)) is used to get the necessary motion of the robot end-effector:

$$\Delta H_{G_k} = {}^{G_k}H_{G_{k+1}} = {}^{G}\hat{H}_{C} \cdot \Delta H_{Ck} \cdot {}^{C}\hat{H}_{G} \qquad (4)$$

[2] Where the usual notation ${}^{A}H_{B}$ means for the homogeneous matrix that transforms coordinate system $\{A\}$ into coordinate system $\{B\}$.

 ISBN: 0 08 043248 4

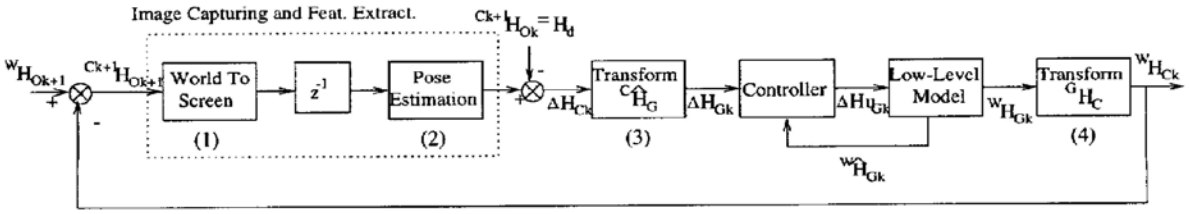

Fig. 3. Complete 6-DOF model.

This resulting matrix is the input to the controller module (to be described later). The controller output is given as an input to the low-level subsystem, the output of which is the current gripper pose. Finally, the motion of the robot implies motion of the camera, according to the true camera-gripper transformation.

The sampling interval of this higher-level model is $T_2 = 8 \cdot T_1$, and, hence, the whole system has a multi-rate structure.

It is worth pointing out that modules (1) and (2) in figure 3 are ideally complementary. In the *World-To-Screen* transformation the real internal parameters of the camera are involved, but in the *Pose-Estimation* module, the estimated values of these parameters are used instead. This estimation was made off-line, using a camera-calibration algorithm (Tsai, 1987), (Sorensen, 1994). Other factors that can disturb the pose estimation are the precision in the estimation of the feature positions in the image plane, the accuracy in the geometric model of the object, the relative speed between the object and the camera, etc.

3. FEATURE EXTRACTION AND POSE ESTIMATION

It is necessary to select some distinguishing visual features of the object to be tracked. In our case, the object used in the experiments was a planar template made of 25 circular, black spots, arranged in matrix form, on a white background. The centers of the spots are the features to be extracted. The position of these centers must be accurately known respect to the local coordinate system placed on the object, $\{O\}$, and will be denoted by: p_w^i, with $i = 1..n$, where n is the number of feature points. The corresponding projections onto the image plane, P_f^i, must be obtained through image analysis.

To avoid the blurring effect in the grabbed image, due to eventual high speeds in the relative camera-object motion, only one of the two fields in each interlaced image is used.

Two different methods have been implemented in order to get the relative orientation and position between the camera and the object of interest.

The first one is a modification of Tsai's calibration method (Tsai, 1987) due to Søresen (Sorensen, 1994). The application of the method gives an estimation of the relative object-camera rotation and translation, matrix R and vector T, respectively.

The second method essayed was originally due to Heuring and Murray (Heuring and Murray, 1996*b*) and (Heuring and Murray, 1996*a*). The main disadvantage respect to Tsai's method is that the rotation matrix is obtained in an incremental way (δR instead R). On the other hand it gives rise to a simpler formulation and more equations are obtained for the same number of points.

Heuring-Murray's method, by construction, introduces an error in the estimation of the rotation parameters. This error is larger as larger are the rotation angles from the reference orientation. After a first estimation of matrix δR, we have angles close to the real ones, if we modify the reference matrix with this first solution, and apply again the algorithm, a better approximation will be obtained. According to this, an iterative scheme has been chosen to estimate the current object-camera orientation. This iterative process should be finished when the incremental rotation matrix reaches the identity matrix. In our implementation, due to the real-time requirements of the system, the number of iterations has been limited to 4, which has been shown to give enough precision in normal conditions.

4. CONTROLLER DESIGN AND EXPERIMENTS

4.1 *Coordinate Decomposition for Control*

The more usual way to control robot manipulators is at low level, in joint coordinates. In our implementation, there is not direct access to the servomotors. Then, it is necessary to program the control laws at a higher level, in cartesian coordinates, and at a relatively large sampling period: $T_1 = 0.028s$.

For implementation of the *Controller* module in figure 3, the transformation matrix must be decomposed in six parameters, that fully determine it: three for translation and three for rotation.

 ISBN: 0 08 043248 4

If we denote by x each one of the six coordinates, and x^u the output of the controller for the corresponding coordinate, x, the following algebraic expresion implements a general discrete controller:

$$x_k^u = a_1\, x_{k-1}^u + \cdots + a_n\, x_{k-n}^u + b_0\, x_k + \cdots + b_m\, x_{k-m} \qquad (5)$$

For the aditivity property being more consistent with the variables used, the coordinates chosen should be refered to a stationary reference system. Then, before decomposing matrix ΔH_{G_k}, it should be transformed into $\Delta H_{G_k}^{(W)}$, which represents the same transformation but refered to the set of axes of the world coordinate system $\{W\}$, the rotation and translation of which can be obtained as:

$$\Delta R_{G_k}^{(W)} = {}^{W}\hat{R}_{G_k} \cdot \Delta R_{G_k} \cdot {}^{G_k}\hat{R}_W \qquad (6)$$
$$\Delta T_{G_k}^{(W)} = {}^{W}\hat{R}_{G_k} \cdot \Delta T_{G_k}$$

Obviously, in the case of the **translation parameters**, the three components of the translation vector are chosen.

The inmediate selection for the **rotation parameters** could be Euler's angles representation of the $\Delta R_{G_k}^{(W)}$ matrix. However, another parameterization of the rotation matrix has been chosen in order to avoid the strong cross-coupling showed by Euler's angles, and to fullfil the aditivity property required in the control variables.

The parameterization for the rotation matrix finally used is the axis-angle specification, $(\bar{k}, \theta) \leftarrow R$, where $\bar{k}$ is the unitary vector on the rotation axis, and θ is the angle rotated about this axis, in radians.

The additivity property of this rotation parameters derives from Varignon's theorem, known from kinematics theory, which has to do with the combination of simultaneous angular velocities in continuous time:

- **Varignon's Theorem**

The following relation:

$$\sum_i (\bar{\omega}_i \wedge \bar{O_i P}) = \bar{\omega} \wedge \bar{OP}$$

(where: $\bar{\omega} = \sum_i \bar{\omega}_i$, O_i: point on the corresponding rotation axis, P: feature point), holds when all the rotation axes intersect in one point O.

In our discrete-time case, if a rotation matrix, ΔR_{G_k}, must be applied in cycle k, the corresponding average angular velocity during this cycle is:

$$\bar{\omega}_k = \frac{\theta}{T_2}\bar{k},$$

Where: $(\bar{k}, \theta) \leftarrow \Delta R_{G_k}$, and T_2 is the sampling period.

If the components of $\bar{\omega}_k$ are used as control variables (x in equation (5) can be replaced by $\bar{\omega}$). Hence, angular velocities are used whose rotation axes converge in the origin of the current $\{G_k\}$ coordinate system, and are acting simultaneously to obtain the $\bar{\omega}_k^u$ as a linear combination of them. The rotation matrix corresponding to the output of the controller is:

$$\Delta R u_{G_k}^{(W)} \leftarrow (\bar{k}, \theta) \leftarrow \bar{\omega}_k^u \cdot T_2$$

As a conclusion, the control variables can be considered as independent variables and will be put together in a row vector, in the form:

$$Q_k = [\frac{\Delta T_{G_k}^{(W)}}{T_2}; \bar{\omega}_k]; \quad \bar{\omega}_k = \bar{k} \cdot \theta \leftarrow \Delta R_{G_k}^{(W)}$$

The control law (the general form of which is in equation (5), replacing x by Q) gives Q_k^u as an output. To come back again to homogeneous matrix formulation:

$$Q_k^u(1:3) \rightarrow \Delta T u_{G_k}^{(W)}$$
$$Q_k^u(4:6) \rightarrow \Delta R u_{G_k}^{(W)}$$

Finally, the converse transformation to (6) is made:

$$\Delta R u_{G_k} = {}^{G_k}\hat{R}_W \cdot \Delta R u_{G_k}^{(W)} \cdot {}^{W}\hat{R}_{G_k}$$
$$\Delta T u_{G_k} = {}^{G_k}\hat{R}_W \cdot \Delta T u_{G_k}^{(W)}$$

4.2 *Controller Design*

As the response of each one of the 6 chosen control variables is close to the response of the one-dimensional model of our visual tracking system, this parameterization of the transformation matrices will allow to develope the control design stage based on the 1-DOF model, and, afterwards, on the 6-DOF system, make some minor modifications in the controller parameters for the final tuning of the system response.

Then, the design of the controllers has been carried out looking for the best performance of the one-dimensional model.

This one-dimensional model was simplified as detailed in (Vargas *et al.*, 1998), in order to make easier the control design stage. The approximated one-dimensional, single-rate block diagram is shown in figure 4. In this simplification it has been assumed that the pair of modules (1) and (2) in figure 3 are exactly complementary, and the same holds for modules (3) and (4).

 ISBN: 0 08 043248 4

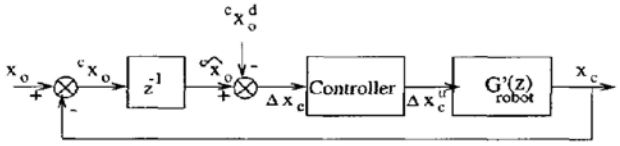

Fig. 4. Approximate model used for controller design.

$G'_{robot}(z)$ in the figure is:

$$G'_{robot}(z) = \frac{7z+1}{z(z-1)}$$

In general, the performance of the controller designed using the 1-DOF model, degrades when applied to the 6-DOF system. The reason is not strictly the unsuitability of the controller extrapolation, but the poorer object's pose estimation achievable in 6-DOF. Another factor that contributes to deteriorate the response in the 6-DOF case is the existing uncertainty in the camera-gripper transformation matrix, ${}^{C}\hat{H}_{G}$, that can slightly deflect the camera motion from the exact desired transformation.

4.3 *Control Experiments*

In the previously mentioned paper, (Vargas *et al.*, 1998), only one or two dimensional experiments were included. In the present work, step responses when all the 6 DOF are controlled will be shown, using both pose-estimation algorithms refered in section 3.

Proportional Controller

The first controller essayed was the proportional controller, which the other ones will be compared with. In figure 5 appears the experimental response of the 6-DOF system, under step input in translation (upper plots) and rotation (lower plots) coordinates, and for both pose-estimation methods.

Pole-Placement Controller

The second controller tested in our visual servoing system was a discrete controller with the general structure:

$$\frac{q_0 + q_1\, z^{-1} + q_2\, z^{-2}}{1 + p_1\, z^{-1} + p_2\, z^{-2}} \qquad (7)$$

designed using the pole-placement method. The specified poles for the closed-loop system were placed in: $p_0 = 0.15$, $(p_1, p_2) = 0.36 \pm 0.15i$, $(p_3, p_4) = 0.17 \pm 0.175i$.

Smith's Predictor Controller

Other controller included for comparison in this work is made of a proportional controller with a Smith's predictor.

The transfer function of this controller is of the form:

$$\frac{k(1-z^{-1})}{(1+ak) + (bk-1)\, z^{-1} - ak\, z^{-2} - bk\, z^{-3}} \qquad (8)$$

with: $a = 7$, $b = 1$ *and* $k = 0.08$.

The most significant improvement provided by the use of this controller, can be apreciated in figure 6, where its step response is compared with the two other controllers, in the case of Tsai's pose-estimation method being used and steps inputs only in the three translation components. Several other experimental tests were made, comparing the behavior of the Smith's predictor controller (also for the pole-placement controller) and the proportional one, but no meaningful differencies were noticed.

5. IMPLEMENTATION

The visual servoing system implementation described in this paper, is made of the following main components:

- *UNIMATION robot system*: Composed by the VAL II controller module and the PUMA 560 robotic arm.
- *Pulnix monocrome, CCD camera.*
- *Personal Computer.* Pentium 160MHz. Plugged into the PC, there is a multiple Digital Signal Processors Motherboard, with three DSP modules, one of which is an image adquisition module. The parallelizing capabilities of this system have not been exploited yet.

The physical link between the personal computer and the robot controller is the serial line using the protocol RS-232C.

6. CONCLUSIONS

This paper introduces a model for a visual servoing system, using the position-based approach. The system is based on the PUMA 560 robot manipulator, operated using the ALTER facility. When operating in ALTER mode, the robot is commanded in cartesian positions; thus, it is not necessary to solve the inverse kinematics problem, and no direct control on the servomotors is available.

The obtained model (which happens to be a multi-rate one) has been validated in some experimental tests, comparing its response to the real response of the system.

Then, several controllers have been designed, starting from an approximated single-rate simplified model. The step response of the controllers with best performance is showed.

 ISBN: 0 08 043248 4

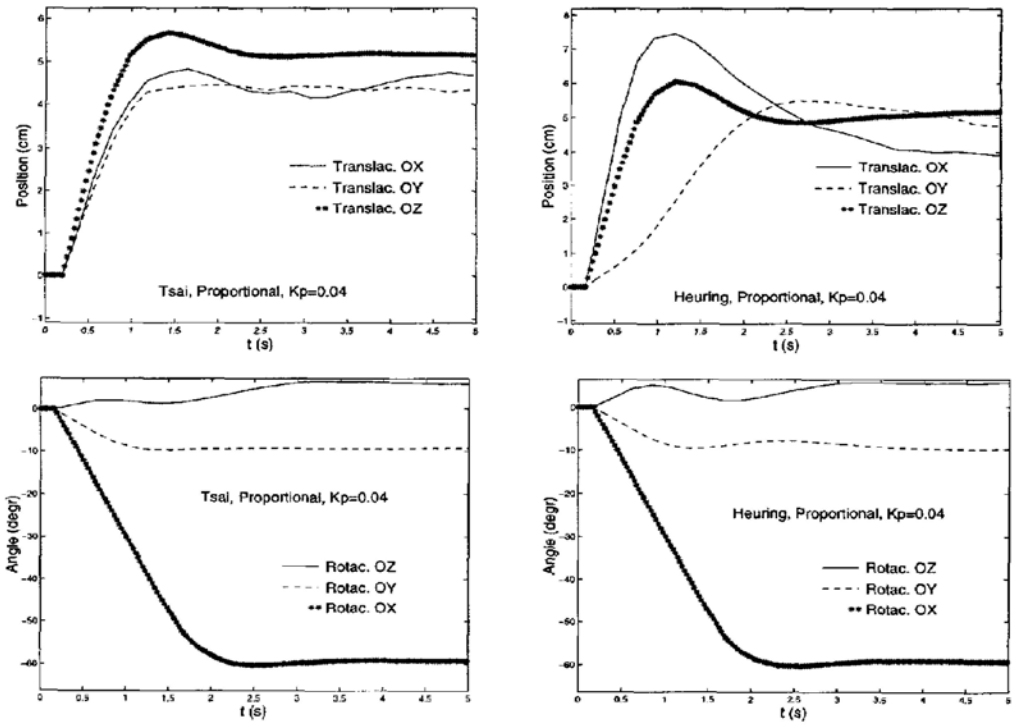

Fig. 5. Response to step input under proportional control

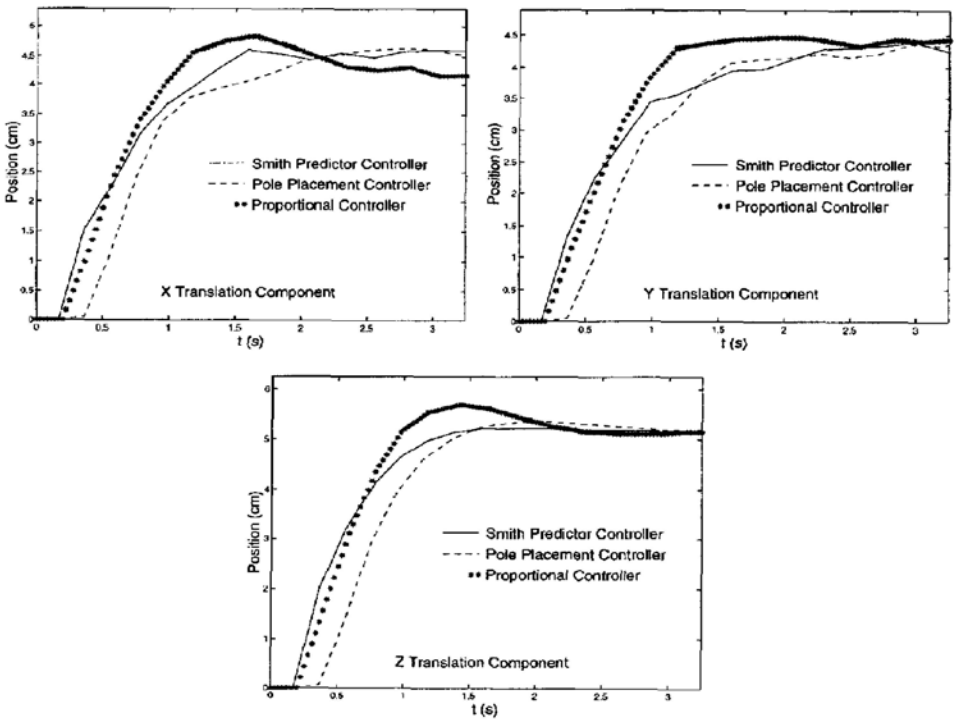

Fig. 6. Comparison of the response in each translation component

 ISBN: 0 08 043248 4

Current and future work is focused on the generalization of the kind of features to be extracted for the object pose estimation. Besides this, a similar visual servoing system is being implemented on a mobile robot.

7. REFERENCES

Corke, P.I. (1996). *Visual Control of Robots. Research Studies.* Press Ltd.

Hashimoto, K. (1993). *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback.* World Scientific.

Heuring, J.J. and D.W. Murray (1996*a*). Slaving head and eye movements for visual telepresence. *British Machine Vision Conference, BMVC'96.*

Heuring, J.J. and D.W. Murray (1996*b*). Visual head tracking and slaving for visual telepresence. *Proc. IEEE Int. Conf. on Robotics and Automation.*

Sorensen, A.T. (1994). *Analysis of Real-Time Vision in Control and Automation of Systems.* Vol. Ph.D. Thesis. Dept. of Automation, IAU, Denmark Technical University.

Tsai, R.Y. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Tran. on Robotics and Automation.*

Tsai, R.Y. and R.K. Lenz (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Tran. on Robotics and Automation.*

Unimation (1986). *Unimate Industrial Robot. User's Guide to VAL II.* Unimation.

Vargas, M., A.R. Malpesa and F.R. Rubio (1998). Modelling and control of a visual servoing system. *Proceedings of the Third European Robotics, Intelligent Systems and Control Conference, EURISCON'98.*

 ISBN: 0 08 043248 4