ABCM Symposium Series in Mechatronics - Vol. 4 - pp.648-657

COMPARISON BETWEEN LOOK-AND-MOVE AND VISUAL SERVO CONTROL USING SIFT TRANSFORMS IN EYE-IN-HAND MANIPULATOR SYSTEM

Ilana Nigri, ilanigri@gmail.com
Marco Antonio Meggiolaro, meggi@puc-rio.br
Raul Queiroz Feitosa, raul@ele.puc-rio.br
Pontifical Catholic University of Rio de Janeiro, Rua Marquês de São Vicente 225 Gávea, Rio de Janeiro, RJ, Brazil

Abstract. *The present work has the objective to develop and implement visual control techniques to self-localize and position robotic manipulators. It is assumed that a monocular camera is attached to the robot end-effector (eye-in-hand configuration). Two classical visual control techniques are studied: look-and-move and visual servo control. The main contribution of this work is the use of the SIFT transform in these control techniques to obtain and correlate key-points between reference images and images captured in real time by the robot camera. The proposed methodology is experimentally validated using a three degree-of-freedom manipulator, with 2 prismatic joints and 1 rotational joint, especially designed and built for this work. The manipulator design is based on the mechanical structure of an x-y-θ coordinate table, powered by DC gearmotors with encoders, controlled by a computer through an electronic system, including control software where all presented methodologies are implemented. A monocular camera fixed to the robot end-effector is able to capture images from the environment, used to control the relative position between the manipulator and a generic object. Preliminary tests are performed on simple circular-shaped objects, without the need for SIFT transforms. It is shown that it is possible to automatically control the relative position between the robot end-effector and the tested circular objects. Next, tests are performed with a photo of an actual manifold panel typically used in submarine interventions. The localization of the manipulator with respect to the panel and its valves is performed with the aid of the SIFT transform, used to determine key-points in the camera images. The experimental results show that the frequency of the visual servo control is very much dependent on the image processing time, as opposed to the look-and-move technique, which only needs to capture a single image. On the other hand, visual servo control presents smaller steady-state positioning errors, because the captured images when the manipulator is close to its desired position are used in the system feedback to improve in real time the robot accuracy.*

Keywords: *robotic manipulator, eye-in-hand, look-and-move control, visual servo control, SIFT transform*

1. INTRODUCTION

Recent developments in robotics and computer vision have allowed their widespread use in several areas. Robotic systems are able to replace humans in dangerous or repetitive tasks, while computer vision allows such robots to recognize objects, shapes and colors, using this information to execute complex tasks.

One application of great interest is based on the use of computer vision to calibrate and self-localize a robotic manipulator. This application can be useful in submarine interventions, where a robotic manipulator is mounted on a Remote Operated Vehicle (ROV) to execute tasks at high depths, such as handling manifold valves. Such task is currently performed by tele-operators. To partially automate this task, the robot must be able to measure in real time its pose with respect to the serviced equipment and its components (such as manifold valves). An application example is based on the TA-40 robotic manipulator, used by Petrobras in submarine interventions.

Several works were presented in the literature, combining robotics with computer vision. The most common application consists on a robot executing a task commanded by visual information.

Inoue & Shirai (1971) built a robotic manipulator with 7 degrees of freedom, with an eye-in-hand system. The objective was to fit an object in a hole of the same format. With their own software, the control estimated the distance between the robot and the object, and it reached the object only using image information.

Allota & Colombo (1999) designed a robot eye-in-hand system. Visual features were obtained through edge-finding. Using a 2D/3D control, the system was able to perform positioning tasks.

Houshangi (1990) designed a system with a fixed camera, and developed a system to capture moving objects.

The present work has the objective to develop and implement visual control techniques to self-localize and position robotic manipulators. It is assumed that a monocular camera is attached to the robot end-effector (eye-in-hand configuration). Two classical visual control techniques are studied: look-and-move and visual servo control. Their main difference is related to the adopted feedback sensors. The first technique uses position sensors with the aid of a single image captured at the beginning of the robot movement. The second technique, on the other hand, does not make use of position sensors, it only relies on several images captured in real time during the robot movement.

Each of these techniques can be implemented according to two different choices for state variables: variables based on pose (positions and orientations), or variables based on image features. When dealing with pose variables, a desired relative pose between the camera and an object is chosen; the robot is then controlled until such position and orientation is achieved. When dealing with image features, the robot only receives an image associated with the desired position, while the control moves the manipulator until its end-effector camera captures an image as similar as possible to the provided one.

In this work, the SIFT transform (Lowe, 1999) is used in these control techniques to obtain and correlate key-points between reference images and images captured in real time by the robot camera. The SIFT transform is robust to rotations, translations, scale and lighting changes, improving the control system robustness.

2. ANALYTICAL BACKGROUND

Next, some techniques used in this work will be presented, necessary to understand the experimental procedure.

2.1. SIFT (Scale Invariant Features Transform)

The main objective of the SIFT algorithm (Lowe, 2004) is the invariant features extraction from images, in order to find matching points between two images. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

The result of the SIFT method applied to an image can be seen in Fig. 1, where the keypoints found by the algorithm are shown in blue, such as the point “orientation” as defined by the algorithm.

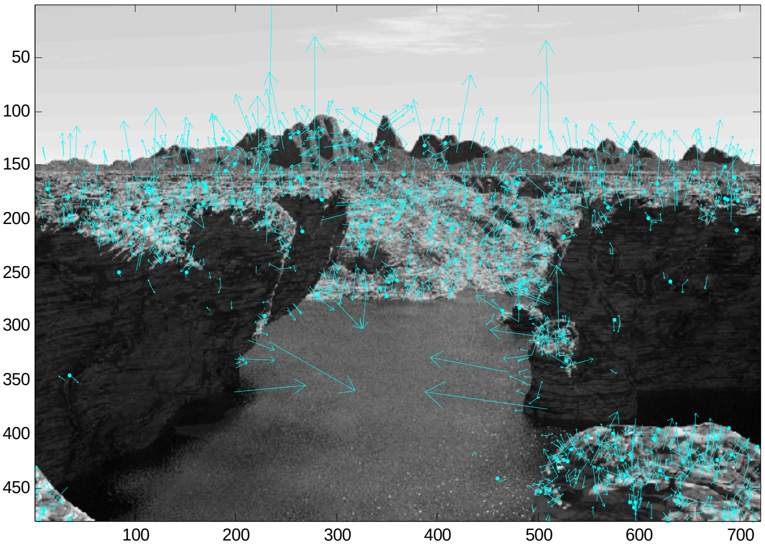

Figure 1 - The result of the SIFT method applied in a image

After finding the keypoints of the image pair, the matching process starts. The correlation between the images is then found. The main objective is to find the same point in the different views of the object. A matching result is shown in Fig. 2.

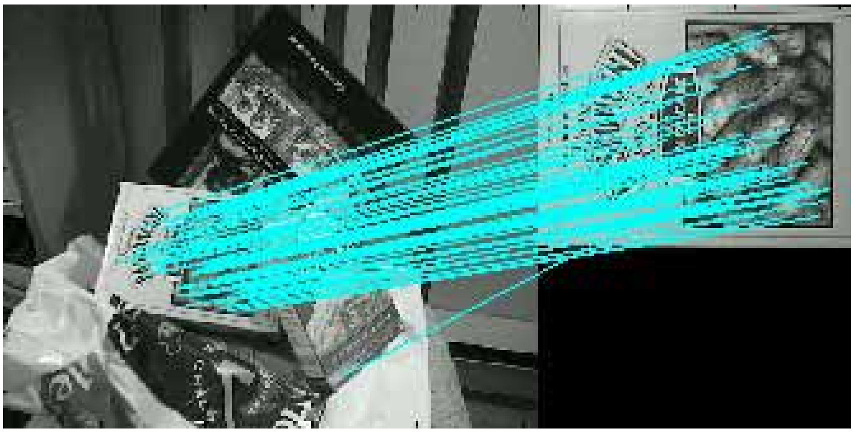

Figure 2 – Matching result between two images

2.2. Control Architecture

The two control techniques based on images, which will be studied in this work, differ each other with respect to the system feedback. Sanderson and Weiss (1980) introduced two concepts to classify visual-servo systems. The look-and-move system uses visual computation to generate the set-points to the joints from a single image, without visual feedback. On the other hand, visual-servo systems do not make use of conventional position controller/sensors, using instead images to correct for the joints errors.

Each of these techniques can be implemented according to two different choices for state variables: variables based on pose (positions and orientations), or variables based on image features. When dealing with pose variables, a desired relative pose between the camera and an object is chosen; the robot is then controlled until such position and orientation is achieved. When dealing with image features, the robot only receives an image associated with the desired position, while the control moves the manipulator until its end-effector camera captures an image as similar as possible to the provided one. The Jacobian matrix is responsible for converting the desired and actual features in inputs to the controller.

Knowing that visual control can be classified from the presence or absence of a conventional position controller, and by the desired variable (pose or image), four different controllers can be defined: look-and-move based on pose, look-and-move based on image, visual-servo based on pose and visual-servo based on image. In the look-and-move control, the system feedback is realized in the joints, using position sensors in the system feedback. Visual-servo control does not use position controllers in the joints. Feedback is realized through images captured in real time. At each control loop, a new image frame is captured and a new difference between the real and desired position is obtained.

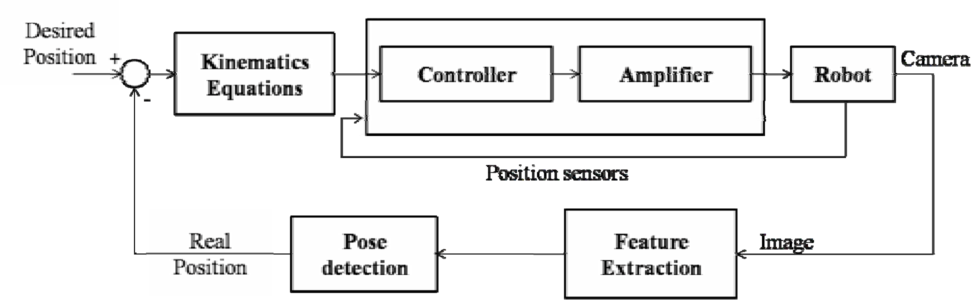

Figure 3. Look-and-Move control based on pose

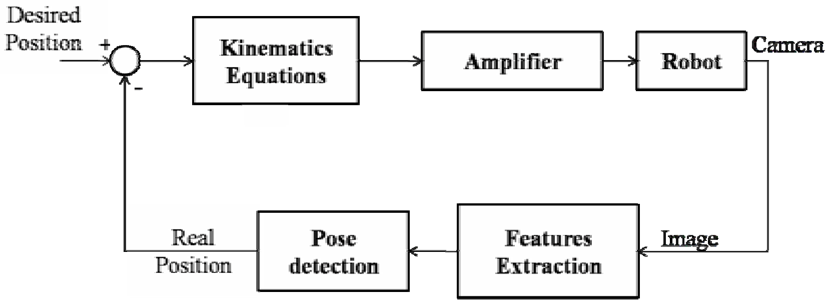

Figure 4. Visual Servo control based on pose

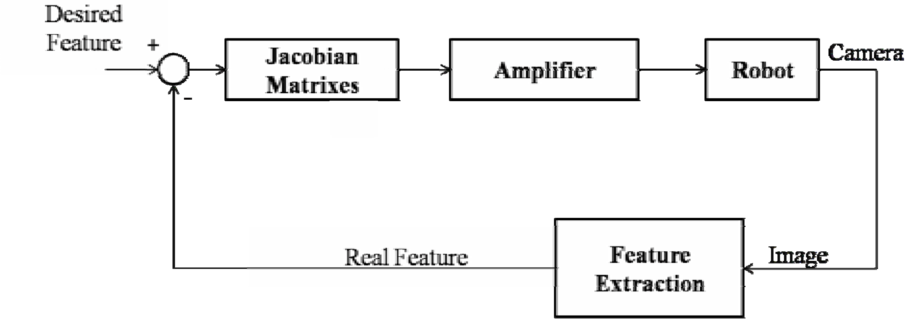

Figure 5. Visual Servo control based on image

2.3. PID Control

Once the desired position is determined, a position control is necessary to transform the desired position into information to the motors. Many techniques can be found in the literature, but for this work PID control was chosen, one of the most common control techniques. The PID control can be understood as a combination of three different techniques: Proportional, Integral and Derivative, following the equation

$$u_i = K_{Pi}\, e + K_{Ii} \int_0^t e\, dT + K_{Di}\, \dot{e} \tag{1}$$

where i represents the link number, u is the resulting force or torque to be applied by each actuator, e is the error between the real and desired position, K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain. The gains were calibrated from experimental procedures.

3. EXPERIMENTAL SYSTEM

The project scheme and main steps are presented in Fig. 6.

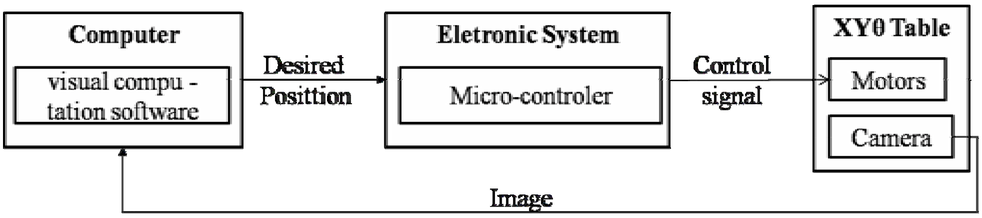

Figure 6. Schematic of the experimental system

The proposed methodology is experimentally validated using a three degree-of-freedom manipulator, implemented using an automated x-y-θ coordinate table. A camera is fixed at the end-effector of the table, extracting an image from the target. Vision software computes the desired coordinates from the target and sends them to an electronic system. A microcontroller inside the electronic system estimates the necessary torques to reach the desired position, and sends it to the coordinate table motors.

3.1. Mechanical System

The automated coordinate table used in this work has 2 prismatic joints and 1 rotational joint, powered by DC gearmotors with encoders. A monocular camera fixed to the robot end-effector is able to capture images from the environment, used to control the relative position between the manipulator end-effector and a generic object. An image of the coordinate table is shown in Fig. 7.

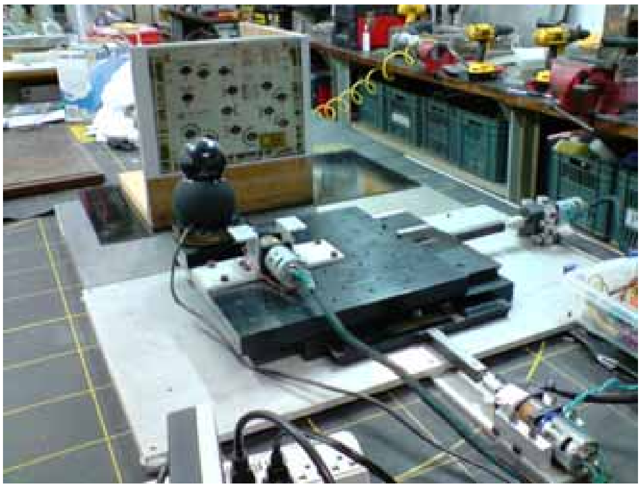

Figure 7. x-y-θ coordinate table

3.2. Electronic System

An electronic system interface has been developed to control the movement of the coordinate table motors using a computer. The electronic system communicates with the computer through a serial port. It contains a microcontroller responsible to execute a PID algorithm and determine the currents to be applied to the motors. The output signals are of the PPM type (Pulse Position Modulation). To activate the motors, speed controllers Banebots BB12-45 are used, which can provide 12A continuous and 45A peaks.

For the controls based on pose, this electronic system receives the information about a desired position from a computer, compares it to the measured positions from the motor encoders, and then calculates and sends signals to generate torques to the motors according to a PID control law. For the image-based controls, the computer is responsible for calculating the errors between the desired image and the captured one, directly sending the control signals to the electronic system, which then acts only converting them to the PPM format.

3.3. Control Software

The Matlab software is chosen to implement the control, because of its comprehensive library on image processing, in addition to its simple-to-use communication with a serial port. The main screen presents a few buttons that allow the user to choose the desired variable to be controlled between pose or image, and among the four control combinations (Look-and-Move based on pose, Look-and-Move based on image, Visual-Servo based on pose and Visual-Servo based on image).

Two types of targets are used in the experiments, one based on a simple circular object, easily identifiable by the image processing software, and another based on a generic 2D image, which requires the use of SIFT. Both are described next.

3.3.1. Circular target

To determine the relative distance between a circular-shaped object and the camera, a few equations had to be developed based on geometric principles. Figure 8 shows a scheme of the experiment using a (red) disc as a target.

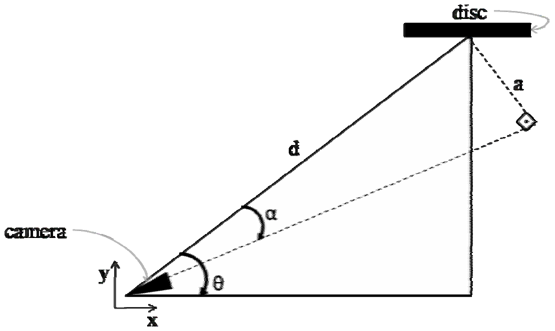

Figure 8. Experiment schematic using a red circular target

The schematic presented in Fig. 8 assumes that the disc axis of symmetry is aligned with the y direction, while x and y represent the coordinate axes from the experimental table. The angles θ is defined in the figure as the angle between the line joining the camera center and the disc center and the x axis, while the α angle is related to the optical axis of the camera. The distance d between the camera and the disc center is also shown. The last parameter a represents the distance between the disc center and the optical axis of the camera. The following equations can then be written

$$x = d\cos\theta \tag{2}$$

$$y = d\sin\theta \tag{3}$$

$$\sin\alpha = \frac{a}{d} \tag{4}$$

$$\theta' = \theta - \alpha \tag{5}$$

When the circle is not centered in the image, it is possible to observe two different values for i_r and i_R, the smaller and the larger semi-axes in the image, respectively. A distance i_a between the image and the disc center can be also observed.

Assuming that the camera and the disc centers are always at the same vertical level, it is possible to affirm that the largest semi-axis will only change its size when the distance in y is changed. In other words, when the camera gets closer to the object, the larger semi-axis will become larger in the image, resulting in

$$d = \frac{K}{i_R} \tag{6}$$

$$\frac{a}{r} = \frac{i_a}{i_R} \tag{7}$$

where K is constant and r is the disc actual radius.

To find the disc rotation with respect to the camera, it is possible to use the ratio between the semi-axes (see Fig. 9):

$$i_r = i_R \sin\theta \Rightarrow \theta = \sin^{-1}\left(\frac{i_r}{i_R}\right) \tag{8}$$

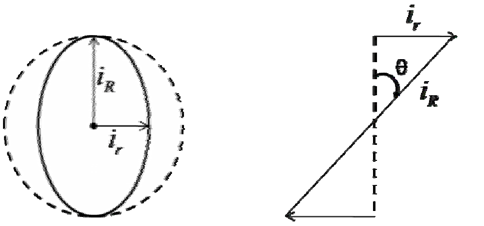

Figure 9. Frontal and upper views from the disc

Once the equations above are determined, it is possible to find the values for x, y and θ'. For the techniques based on image, it is necessary to write the equations for the feature variables $\$_1$, $\$_2$ and $\$_3$. For this work, these variables will be based on i_a, i_r and i_R, and are defined as: $\$_1 = \frac{1}{i_R}$, $\$_2 = \frac{i_r}{i_R}$, $\$_3 = \frac{i_a}{i_R}$. Rewriting all the equations, the values of x, y and θ' become

$$x = K\,\$_1\sqrt{1-\$_2^2} \tag{9}$$

$$y = K\,\$_1\,\$_2 \tag{10}$$

$$\theta' = \sin^{-1}(\$_2) - \sin^{-1}\left(\frac{r\,\$_3}{K\,\$_1}\right) \tag{11}$$

It is possible now to write in matrix form the relationship between the position vector q and the feature vector $\$$. From by the parameter vector p defined below it is possible to find the transform matrices $J_{\$p}$ and J_{qp}:

$$\underbrace{\begin{pmatrix}\delta\$_1\\ \delta\$_2\\ \delta\$_3\end{pmatrix}}_{\delta\$} = J_{\$p}\underbrace{\begin{pmatrix}\delta d\\ \delta\theta\\ \delta a\end{pmatrix}}_{Sp} \quad\text{and}\quad \underbrace{\begin{pmatrix}\delta x\\ \delta y\\ \delta\theta'\end{pmatrix}}_{\delta q} = J_{qp}\underbrace{\begin{pmatrix}\delta d\\ \delta\theta\\ \delta a\end{pmatrix}}_{\delta p} \tag{12}$$

$$\begin{pmatrix} \delta x \\ \delta y \\ \delta\theta' \end{pmatrix} = J_{qp}\, J_{\$p}{}^{-1} \begin{pmatrix} \delta\$_1 \\ \delta\$_2 \\ \delta\$_3 \end{pmatrix} \tag{13}$$

where,

$$J_{\$p}{}^{-1} = \begin{bmatrix} K & 0 & 0 \\ 0 & \sec\theta & 0 \\ 0 & 0 & r \end{bmatrix} \rightarrow \begin{bmatrix} K & 0 & 0 \\ 0 & \dfrac{1}{\sqrt{1-\$_2^2}} & 0 \\ 0 & 0 & r \end{bmatrix} \tag{14}$$

$$J_{qp} = \begin{bmatrix} \sin\theta & d\cos\theta & 0 \\ \cos\theta & -d\sin\theta & 0 \\ \dfrac{1}{d}\tan(\theta-\theta') & 1 & -\dfrac{1}{d}\sec(\theta-\theta') \end{bmatrix} = \begin{bmatrix} \sqrt{1-\$_2^2} & -K\,\$_1\$_2 & 0 \\ \$_2 & K\,\$_1\sqrt{1-\$_2^2} & 0 \\ \dfrac{r\,\$_3}{K\,\$_1\sqrt{K^2\$_1^2-r^2\$_3^2}} & 1 & -\dfrac{1}{\sqrt{K^2\$_1^2-r^2\$_3^2}} \end{bmatrix} \tag{15}$$

Knowing the real and desired values of $\$_1$, $\$_2$ and $\$_3$, it is possible to find δx, δy, $\delta\theta'$ using Eq.13.

3.3.2. Generic 2D Target

For the second part of the experiments, instead of using a red circle, a generic 2D image is chosen as a target. The chosen image reflects the view from a manipulator of a manifold control panel. Using the SIFT method, the keypoints in the images are obtained. First, SIFT is applied to a reference image from a known position of the camera, obtaining the coordinates of the keypoints in space. With these reference coordinates, it is possible to find the coordinates from the same points, found by a matching technique, in images taken from any other position. The experiment schematic is shown in Fig.10, where x represents the actual distance between the keypoints and the center of the control panel and x_i represents the distance in pixel coordinates from their projection to the center of the image.

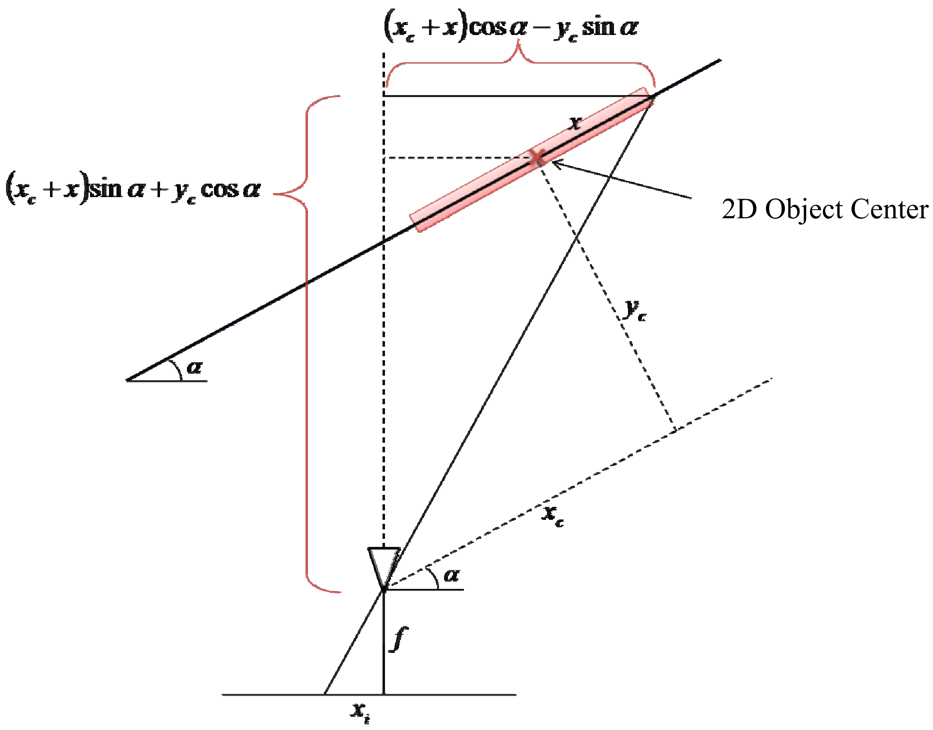
$(x_c + x)\cos\alpha - y_c \sin\alpha$
$(x_c + x)\sin\alpha + y_c \cos\alpha$
x
2D Object Center
y_c
α
x_c
α
f
x_i

Figure 10. Experiment schematic using generic 2D objects as targets

Once the keypoint space and pixel coordinates are determined, and knowing that f is the focal length, one can get

$$\frac{(x_c + x)\sin\alpha + y_c\cos\alpha}{f} = \frac{(x_c + x)\cos\alpha - y_c\sin\alpha}{x_i} \tag{16}$$

and consequently

$$\begin{bmatrix} f & x_i & x \cdot x_i \end{bmatrix} \cdot \underbrace{\begin{pmatrix} y_c \tan\alpha - x_c \\ x_c \tan\alpha + y_c \\ \tan\alpha \end{pmatrix}}_{X} = \begin{bmatrix} f \cdot X \end{bmatrix} \tag{17}$$

For M pairs of points (x, x_i), and using the pseudo-inverse formulation, the X vector can be determined by

$$\underbrace{\begin{bmatrix} f & x_{i1} & x_1 x_{i1} \\ f & x_{i2} & x_2 x_{i2} \\ \vdots & \vdots & \vdots \\ f & x_{im} & x_m x_{im} \end{bmatrix}}_{A} \cdot X = f \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}}_{B} \Rightarrow A \cdot X = B \Rightarrow X = pinv(A) \cdot B \tag{18}$$

Once found the vector X, the desired distances (x_c, y_c, α) can be determined.

4. RESULTS

Two different types of tests were performed. The first tests were performed using a red circle as target, and the second using the image of the manifold panel, representing a real application.

The circle tests used images with 960 x 720 pixels. The panel test, on the other hand, used a lower resolution because the SIFT algorithm is relatively slow, considerably increasing the processing time (5 seconds to process a 960 x 720 pixel image in Matlab). As the visual servo control depends directly on the processing time, the image size was changed to 352 x 288 pixels. In look-and-move control, however, the image could be used in high resolution because this technique only needs to process a single image.

In the first test, it was desired to position the camera 10 cm in the X axis, 10 cm in the Y axis, and with no rotation. In this position, it would be desired to see in the image $i_r = i_R = 195$ pixels, and $i_a = 10$ pixels. The camera starts in x = 0cm, y = 0cm and θ = 0°. The initial image from the camera and the desired image are indicated in Fig.11.

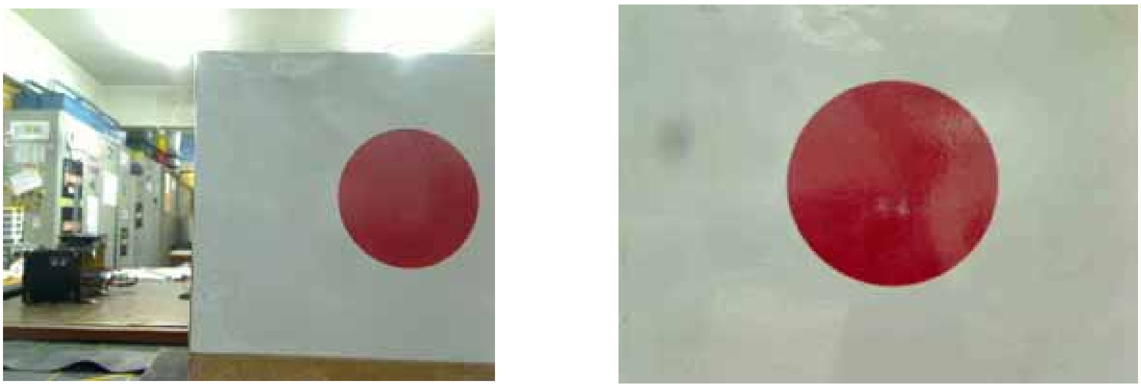

Figure 11. Initial image (left) and final desired image (right)

Table 1 indicates the reached position (x, y, θ) for each control technique. For look-and-move control, the relative position ($x_{rel}, y_{rel}, \theta_{rel}$) found by the software from the first captured image is also indicated. This relative position is not shown for visual servo control, because it is continually changed at each new image that is processed. The last 3 lines represent the final values of the features i_a, i_r and i_R, which can be compared to the desired values 195,195 and 10 discussed above.

Table 1. Final and desired positions for the four control techniques

	Look-and-Move Control based on pose	Look-and-Move Control based on image	Visual-Servo Control based on pose	Visual-Servo Control based on image
x_{rel}	10.1 cm	7.9 cm	48 interactions	20 interactions
y_{rel}	8.0 cm	8.3 cm		
θ_{rel}	0°	3°		
x	10.7 cm	9.0 cm	10.0 cm	5.4 cm
y	8.1 cm	8.4 cm	10.3 cm	10.5 cm
θ	0°	3°	0°	10°
i_r	179 pixels	183 pixels	200 pixels	189 pixels
i_R	180 pixels	183 pixels	202 pixels	195 pixels
i_a	15 pixels	-37 pixels	-20 pixels	-37 pixels

Note from the table that the visual-servo control based on pose obtained the best positioning result. The look-and-move control is not very accurate because it uses a single image taken from the starting position of the manipulator. Note also that the controls based on image features obtained different final positions from the desired ones. This happened because the chosen final configuration was close to a singularity of the image Jacobian matrix. Therefore, the final image captures by the camera was actually very similar from the desired one, however it was taken from a different pose. Further tests with desired final positions away from this singularity resulted in better results for the image-based controls, similar to the pose-based ones.

For the panel test, it was desired to position the camera in x=8cm, y=15cm, and θ=30°. The initial and desired views are presented in Fig.12. The position results are presented in the Table 2.

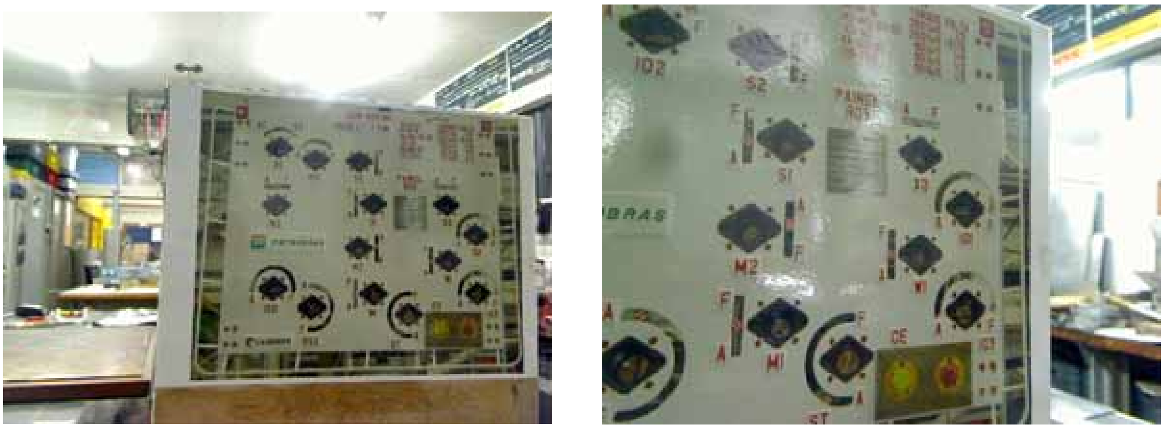

Figure 12. Initial image (left) and final desired image (right)

Table 2. Final and desired positions for the four control techniques

	Look-and-Move Control based on pose	Look-and-Move Control based on image	Visual-Servo Control based on pose	Visual-Servo Control based on image
x_{rel}	6.6 cm	6.6 cm	13 interactions	50 interactions
y_{rel}	15.9 cm	19 cm		
θ_{rel}	26.5°	38°		
x	6.8 cm	6.8 cm	6.5 cm	7.0 cm
y	16.3 cm	19.4 cm	17 cm	16.6 cm
θ	27°	38°	34°	33°

Table 2 shows that for look-and-move control the biggest error source was due to the limitations in the resolution of the image. The relative values x_{rel}, y_{rel} and θ_{rel} estimated from a single image presented significant errors of up to 26%, however the position control was able to move the camera to actual positions x, y and θ within 2% of x_{rel}, y_{rel} and θ_{rel}.

5. CONCLUSIONS

The main objective of this work consisted in comparing the look-and-move and visual-servo image control techniques. Using the SIFT algorithm it was possible to apply the visual control to position the camera in any pose with respect to a generic 2D image. Even though it is very used in the literature, SIFT is still a slow algorithm when using it in real time. With the red circle tests, it was possible to see how visual-servo is better than look-and-move when the processing time is not an issue. Controls based on pose and on image had very similar results, except for configurations close to singularities in the image Jacobian matrix.

6. REFERENCES

Allota, B., & Colombo, C. (1999). On the use of linear camera-object interaction models in visualservoing. IEEE Transacttion on Robotics and Automation , pp. 350-357.

Houshangi, N. (1990). Control of a robotic manipulator to grasp a moving target using vision. IEEE international Conference on Robottics and Automation , pp. 604-609 vol.1.

Inoue, H., & Shirai, Y. (1971). Guiding a robot by visual feedback assembling tasks. Eletrotechnical Laboratory Chiyoda-ku Tokyo, Japan.

Lowe, D. (1999). Object Recognition from Local ScaleInvariant Features. ICCV, Kerkyra.

Sanderson, A., & Weiss, L. (1980). Image-based visual servo control using relational graph error signals. Proc. IEEE, pp. 1074-1077.

7. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.