ON THE USE OF THE SIFT TRANSFORM TO SELF-LOCATE AND POSITION EYE-IN-HAND MANIPULATORS USING VISUAL CONTROL

ILANA NIGRI, RAUL Q. FEITOSA

Department of Electrical Engineering, Pontificial Catholic University of Rio de Janeiro
Rua Marquês de São Vicente 255 Gávea, Rio de Janeiro, RJ, BRAZIL
E-mails: ilanigri@gmail.com, raul@ele.puc-rio.br

MARCO A. MEGGIOLARO

Department of Mechanical Engineering, Pontificial Catholic University of Rio de Janeiro
Rua Marquês de São Vicente 255 Gávea, Rio de Janeiro, RJ, BRAZIL
E-mail: meggi@puc-rio.br

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 Scale Invariant Feature Transform (SIFT) 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 automated coordinate table especially designed and built for this work.

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

Resumo— O presente trabalho tem por objetivo desenvolver e implementar técnicas de controle visual para auto-localizar e posicionar manipuladores robóticos. Assume-se que uma câmera monocular é fixada na extremidade do manipulador. Duas técnicas clássicas de controle visual são estudadas: *look-and-move* e controle servo-visual. A principal contribuição deste trabalho é no uso da transformada SIFT (*Scale Invariant Feature Transform*) nestas técnicas de controle para obter e correlacionar pontos-chave entre imagens de referência e imagens obtidas em tempo real pela câmera do robô. A metodologia proposta é validada experimentalmente usando uma mesa coordenada de 3 graus de liberdade especialmente projetado e construído para este trabalho.

Palavras-chave— manipulador robótico, câmera monocular, controle *look-and-move*, controle servo-visual, transformada SIFT

1 Introduction

One robotic application of great interest is to use computer vision to calibrate and self-localize a robot. This application can be useful e.g. 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 (Augustson, 2007).

Several works were presented in the literature, combining robotics with computer vision (Hartley & Zisserman, 2000; Hutchinson et al., 1996). The most common application consists on a robot executing a task commanded by visual information (Smith & Papanikolopoulos, 1996).

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. By software, the robot successfully estimated its distance from the object only using image information.

Houshangi (1990) designed a system with a fixed camera, which captures moving objects. 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.

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 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 robot until its end-effector camera captures an image as similar as possible to the provided one.

In this work, the SIFT (*Scale Invariant Feature Transform*) is used to obtain and correlate key-points between reference images and images captured in real time by the robot camera (Lowe, 2004). SIFT is robust to rotations, translations, scale and lighting changes, improving the control system robustness.

2 Analytical Background

2.1 SIFT (Scale Invariant Features Transform)

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

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.

2.2 Visual Control Architecture

The two control techniques based on images 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 use several images taken in real time 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 robot until its end-effector camera captures an image as similar as possible to the provided one. Visual-servo systems based on image require the use of an Image Jacobian matrix, which is responsible for converting the errors between the desired and actual features into inputs to the controller. For controls based on pose, feature extraction must also be performed, however there is no need to use an Image Jacobian since the kinematic equations can calculate the system errors from the extracted pose.

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 (Figs. 1-3): 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 may use the position sensor information from the joints, but 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.

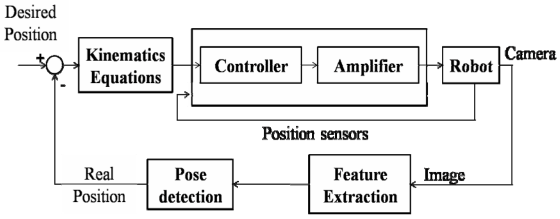

Figure 1. Look-and-Move control based on pose

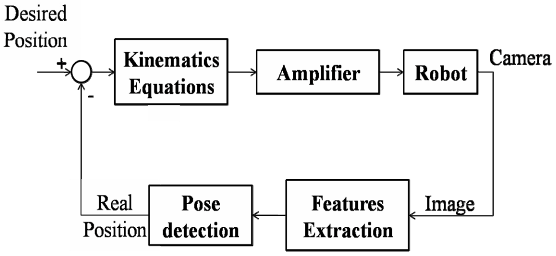

Figure 2. Visual Servo control based on pose

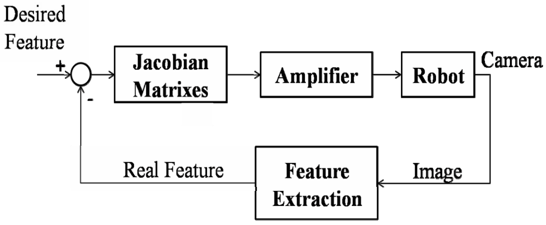

Figure 3. Visual Servo control based on image

2.3 Position Control Architecture

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

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

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 are calibrated from experimental procedures.

3 Experimental System

The project scheme and main steps are presented in Fig. 4.

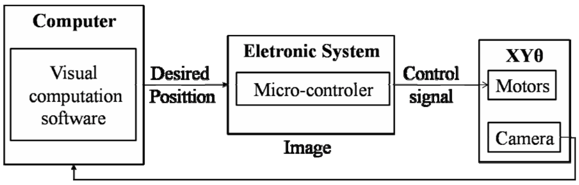

Figure 4. Scheme of the experimental system

The proposed methodology is experimentally validated using a three degree-of-freedom robot, implemented from the automation of an 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 robot end-effector and a generic object. A photo of the coordinate table is shown in Fig. 5. The system kinematic and dynamic models are presented in (Nigri, 2009).

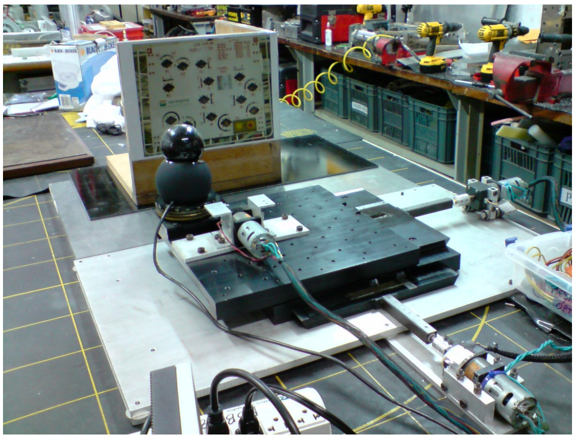

Figure 5. The automated x-y-θ coordinate table

3.2 Electronic System

An electronic system interface has been developed to control the movement of the coordinate table motors using a computer. The 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, Banebots^{TM} speed controllers are used, which can provide 12A continuous currents with 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 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^{TM} software environment is chosen to implement the controls, because of its comprehensive library on image processing, in addition to its simple-to-use communication with a serial port. The main screen from the developed controller interface 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 or Visual-Servo, based on pose or 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 the SIFT transform. Both types are described next.

4 Target-Dependent Formulation

4.1 Circular Target

To determine the relative distance between a circular-shaped object and the camera, a few equations are developed based on geometric principles. Figure 6 shows a scheme of the experiment using a (red) disc as a target.

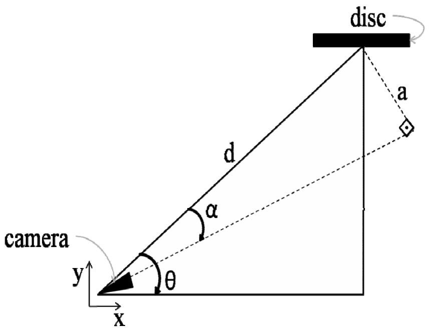

Figure 6. Experiment scheme using a disc as target

The schematic presented in Fig. 6 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 angle 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 from the resulting ellipsis 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 actual radius of the disc. The above equations were obtained from the basic relations of similar triangles.

To find the disc rotation with respect to the camera, it is possible to use the ratio between the semi-axes (see Fig. 7):

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

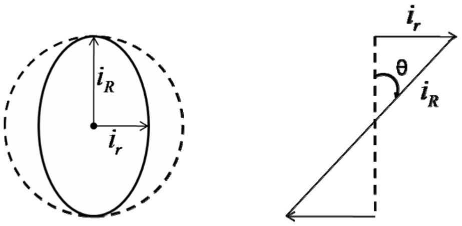

Figure 7. Frontal and upper views from the disc

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, defined as $\$_1 = 1/i_R$, $\$_2 = i_r/i_R$, $\$_3 = i_a/i_R$. 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 the parameter vector ***p*** defined below it is possible to find the Image Jacobian transform matrices $J_{\$p}$ and J_{qp} that correlate small displacements $\delta\$$, δp and δq:

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

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

where

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

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

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

4.2 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 sub-sea 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. 8, 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.

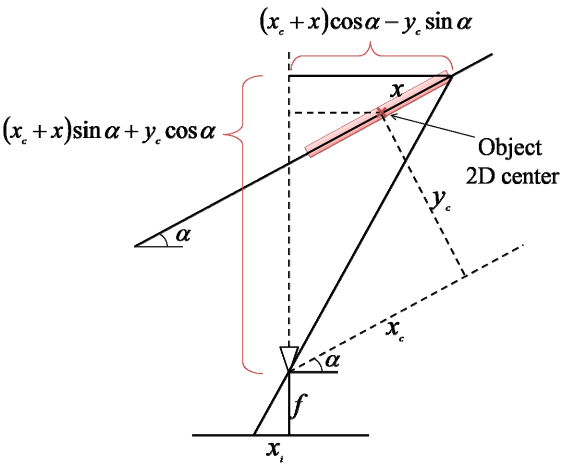
$(x_c + x)\cos\alpha - y_c \sin\alpha$
$(x_c + x)\sin\alpha + y_c \cos\alpha$
x
Object
2D center
y_c
α
x_c
α
f
x_i

Figure 8. Experiment scheme using generic 2D objects as targets

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

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

and consequently

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

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

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

$$A \cdot X = B \Rightarrow X = pinv(A) \cdot B$$

where $pinv(A)$ is the pseudo-inverse of matrix A.

Once the vector $\boldsymbol{X}$ is found, the desired distances (x_c, y_c, α) can be determined.

5 Experimental Results

Two different types of tests are performed. The first tests use a red circle as a target, and the second use the image of the manifold panel, to represent an actual application.

The circle tests use images with 960 x 720 pixels. The panel tests, on the other hand, use 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^{TM}). As the visual servo control depends directly on the processing time, the image size is changed to 352 x 288 pixels. In look-and-move control, however, a high resolution image can be used because this technique only needs to process a single pair of images.

In the circle test, it was desired to position the camera 100 mm in the X axis, 100 mm in the Y axis, and with no rotation with respect to the circle. 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 seen from the camera and the desired image are indicated in Fig. 9.

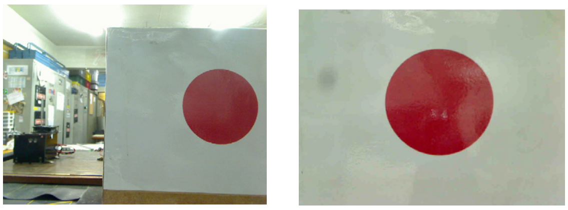

Figure 9. Initial image (left) and final desired image (right)

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}, θ_{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 as each new image 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 pixels 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}	101 mm	79 mm	48 interactions	20 interactions
y_{rel}	80 mm	83 mm		
θ_{rel}	0°	3°		
x	107 mm	90 mm	100 mm	54 mm
y	81 mm	84 mm	103 mm	105 mm
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 visual-servo control based on pose obtained the best positioning results. The look-and-move control is not very accurate because it uses a single image taken from the starting position of the robot. 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 captured by the camera was actually very similar from the desired one, however it was taken from a significantly different pose. Further tests with desired final positions away from this singularity resulted in lower errors for the image-based controls, similar to the pose-based ones.

For the panel test, it was desired to position the camera in x = 80 mm, y = 150 mm, and with θ = 30° with respect to the panel. The initial and desired views are presented in Fig. 10. The position results are presented in the Table 2.

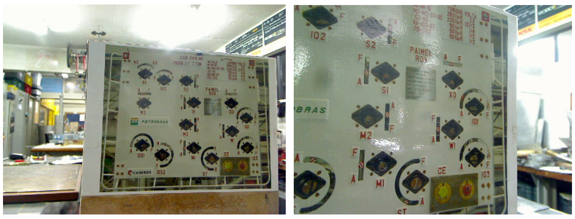

Figure 10. Initial image (left) and final desired image (right)

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}	66 mm	66 mm	13 interactions	50 interactions
y_{rel}	159 mm	190 mm		
θ_{rel}	26.5°	38°		
x	68 mm	68 mm	65 mm	70 mm
y	163 mm	194 mm	170 mm	166 mm
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, not on the position control itself. This is because the relative values x_{rel}, y_{rel} and θ_{rel} estimated from a single image with limited resolution 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}.

6 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 visual control to position the camera in any pose with respect to a generic 2D image. The resulting control is robust because the SIFT technique can handle image translations, rotations, scaling, and even illumination changes. But, even though it is very used in the literature, SIFT is still a slow algorithm when used in real time. The large resulting sampling period may increase too much the time response of the system, or even make it unstable if high PID gains are used. 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.

References

Allota B., Colombo C., “On the use of linear camera-object interaction models in visualservoing,” IEEE Transaction on Robotics and Automation , pp. 350-357, 1999.

Augustson T.M., “Vision based real time calibration of robots with application in subsea interventions,” M.Sc. Thesis, Pontifical Catholic University of Rio de Janeiro, 2007.

Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.

Houshangi N., “Control of a robotic manipulator to grasp a moving target using vision,” IEEE International Conference on Robotics and Automation, pp. 604-609, 1990.

Hutchinson S., Hager G., Corke P., “A tutorial on visual servo control,” IEEE Transactions on Robotics and Automation , pp. 651-670, 1996.

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

Lowe D.G., “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision , v.60, n.2, pp. 91-110, 2004.

Nigri I., “Comparação entre Controles Look-and-Move e Servo-Visual Utilizando Transformadas SIFT em Manipuladores do Tipo Eye-in-Hand,” M.Sc. Thesis, Pontifical Catholic University of Rio de Janeiro, 2009.

Sanderson A., Weiss L., “Image-based visual servo control using relational graph error signals,” Proc. IEEE International Conference on Robotics and Automation, pp. 1074-1077, 1980.

Smith E., Papanikolopoulos N. “Vision-Guided Robotic Grasping: Issues and Experiments,” IEEE International Conference on Robotics and Automation, pp. 3203-3208, 1996.