Binocular visual tracking and grasping of a moving object with a 3D trajectory predictor

J. Fuentes-Pacheco[1], J. Ruiz-Ascencio*[1], J. M. Rendón-Mancha[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico,
Interior Internado Palmira S/N, Col. Palmira, C.P. 62490, Cuernavaca, Morelos, México.
[2] Universidad Autónoma del Estado de Morelos.
Av. Universidad 1001, Col. Chamilpa, C.P. 62209, Cuernavaca, Morelos, México.
*josera@cenidet.edu.mx

ABSTRACT
This paper presents a binocular eye-to-hand visual servoing system that is able to track and grasp a moving object in real time. Linear predictors are employed to estimate the object trajectory in three dimensions and are capable of predicting future positions even if the object is temporarily occluded. For its development we have used a CRS T475 manipulator robot with six degrees of freedom and two fixed cameras in a stereo pair configuration. The system has a client-server architecture and is composed of two main parts: the vision system and the control system. The vision system uses color detection to extract the object from the background and a tracking technique based on search windows and object moments. The control system uses the *RobWork* library to generate the movement instructions and to send them to a C550 controller by means of the serial port. Experimental results are presented to verify the validity and the efficacy of the proposed visual servoing system.

Keywords: linear prediction, visual servoing, tracking, grasping, stereo vision, camera calibration.

RESUMEN
En este trabajo se presenta un sistema servo control visual binocular *eye-to-hand* capaz de seguir y asir un objeto móvil en tiempo real. Se utilizaron predictores lineales para estimar la trayectoria del objeto en tres dimensiones, estos son capaces de predecir futuras posiciones incluso si el objeto es temporalmente ocluido. Para su desarrollo se utiliza un robot manipulador CRS T475 de seis grados de libertad y dos cámaras fijas con una configuración de par estéreo. El sistema tiene una arquitectura cliente-servidor y se compone de dos partes principales: el sistema de visión y el sistema de control. El sistema de visión utiliza una detección por color para extraer el objeto del fondo y una técnica de seguimiento basada en ventanas de búsqueda y momentos del objeto. El sistema de control emplea la librería *RobWork* para generar las instrucciones de movimiento y enviarlas a un controlador C550 por medio del puerto serial. Se presentan resultados experimentales para verificar la validez y eficacia del sistema servo control visual.

Palabras clave: Predicción lineal, servo control visual, seguimiento, visión estéreo, calibración de cámaras.

1. Introduction

At the present time, a great number of robot manipulators have been incorporated into the industrial environment, the service sector (*e.g.*, transport, communications, education, among others), etc. However, the application of these robots to perform a given task depends, in great measure, on a *priori* knowledge of the work environment and the localization of the objects to manipulate. This limitation is due to the fact that most commercial industrial robots do not integrate sensory systems that allow them to adapt to their environment.

Among the most attractive sensory elements to increase the autonomy in robots, artificial vision systems have been highlighted [1, 2, 3]. These have the advantage of being able to emulate the human vision system to provide relevant information on the status of robots and their immediate physical environment. The importance

of having visual information stands out in applications with unstructured or changing environments.

In robotics, the use of visual feedback to coordinate the movements of a robot manipulator receives the name of visual servoing. This term was introduced by Hill and Park [2] in the year 1979. Visual servoing is the result of fusing several areas, among which are mentioned: image processing, kinematics, dynamics, control theory and computing in real time. The task of visual servoing is to control the pose of the end effector of the robot by means of visual information, making it able to manipulate its work environment instead of only to observe [3].

The problem of grasping a target in motion with a robotic manipulator has been shown in different works. Allen *et al*. [4] presented a system to track and grasp an electric toy train moving in an oval path using calibrated static stereo cameras. Nomura et al. [5] proposed a method to grasp efficiently the objects and developed a system able to grasp industrial parts moving on a conveyor belt by controlling a 6DOF robot arm with a camera mounted on its gripper. Sen *et a*l. [6] implemented a real time vision system with a single camera for identifying and intercepting several objects. Ge *et al*. [7] proposed a visual servo system for real-time tracking and grasping of a moving object and a parallel method was adopted to raise matching speed. These researchers have recognized that the main problems in the visual servoing are to solve the delay introduced by image processing or the response of the robot system and resolve the target occlusion. These troubles are the major reason for a limited performance in the tracking and grasping process which can be solved through of the use of predictive algorithms.

The prediction of movement is a very important element that all object tracking algorithms for visual servoing should contain. As in the visual process of the human brain [8], in this work we use prediction as an alternative to compensate for the delay produced between stimulus and response. The linear predictor has been widely studied and used to solve problems of signal processing, time series analysis for economy applications, identification of control problems, among others [9]. In spite of this, the works that employ the linear predictor to estimate the movement of objects have been few.

Yeoh and Abu-Bakar [10] present a tracking algorithm based on a linear prediction of second order solved by the Maximum Entropy Method. It attempts to predict the centroid of the moving object in the next frame, based on several past centroid measurements. Balkenius and Johansson [8] use a prediction module which consists of a linear predictor with the purpose of predicting the location that a moving object will have and thus generate the control signal to move the eyes of a humanoid robot, which is capable of using behavior models similar to those of human infants to track objects. Matas and Zimmermann [11] represent the tracked object as a constellation of spatially localized linear predictors which are trained on a single image sequence. In a learning stage, sets of pixels whose intensities allow for optimal prediction of the transformations are selected as a support for the linear predictor. Ellis and Dowson [12] employ similar linear predictors. They use banks of linear displacement predictors trained online to achieve a fast tracking of arbitrary image features with no prior model. The approach is demonstrated in real time on a number of challenging video sequences and experimentally compared to other tracking approaches.

This paper presents a binocular eye-to-hand visual servoing system that is able to track and grasp a moving object in real time. In the tracking module, we use three linear predictors (one for each

component of the three dimensions) to predict and generate the trajectory that will describe the 3D object position in the near future, therefore, our manipulator robot is able to track and grasp a moving object, even if the object is temporarily occluded. The training of the predictors is done offline and they adapt online to new movements of the object over time. The visual servoing system is implemented with a CRS T475 manipulator robot with six degrees of freedom and two fixed cameras in a stereo pair configuration.

This paper has eight main sections. In Section 2, the major elements to design the eye-to-hand visual servoing system are presented. Section 3 is devoted to the detailed description of the system architecture, and the principle of camera calibration is briefly given in Section 4. In Section 5, the vision system is described. The next section contains a description of the control system. In Section 7, some experimental results are given. Section 8 presents conclusions drawn from this work.

2. Problem definition

It is necessary to consider 3 main elements in the design of a visual servoing system: the selection of the control architecture, the robot-camera configuration and the vision algorithms to achieve the visual feedback. Given what was mentioned above, in this work we identified the following:

The visual information used in the control loop is employed to establish a 3D position-based visual servo [4, 13] system (as opposed to an image-based visual servo [5, 14]), in which the features extracted from the images are used as an intermediate step to estimate the location of the object explicitly (defined in 3D Cartesian space) with respect to the cameras that observe it (see Figure 1).

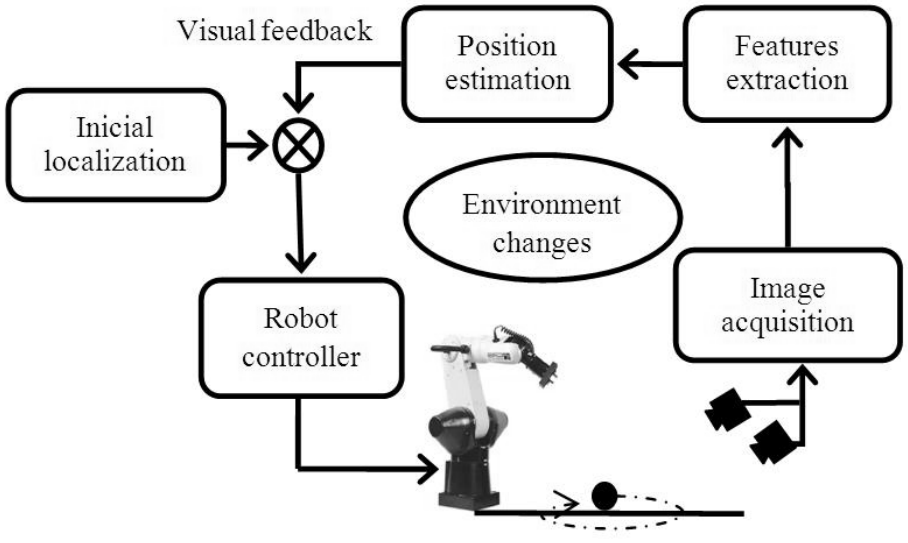


Figure 1. Visual servoing based on the position.

To perform stereo vision, two fixed converging (or toed-in) cameras are used in the binocular eye-to-hand (or stand alone) configuration, positioned in such a way that they capture the robot and its work space. A calibration process is applied to the cameras to establish a correspondence between the coordinates of the image and the robot workspace. The way of placing the optical axes of cameras: parallel or converging; the camera robot configurations: eye-in-hand, eye-to-hand or their combination; and the calibration process are well known in the field of visual servoing and we have used this knowledge [14, 15, 16] to select the appropriate set-up according to the problem and thus design our system.

Because the system should work in real time and in an environment with an object in movement, part of our problem was to give an old and semi-industrial robot real time interactivity. Likewise, the algorithms for processing digital images must be fast, accurate and have low computational cost. In addition, they should be robust to lighting variations and partial or total occlusion of the object.

The work universe consists only of a rigid object of spherical shape and a single color; in many research works [5, 17, 18], the use of these distinctive object features has enabled to achieve a real time tracking speed. The type of movement is a uniform circular trajectory plus a slow bidirectional displacement in the plane of the circle, with low speed. The experimental environment is our Artificial Intelligence Laboratory.

3. The system

The system is divided into two major subsystems: the vision and the control systems. The vision system is responsible for capturing the image sequences obtained by a pair of cameras to extract the visual characteristics of the object and determine its position. The control system receives the 3D coordinates of the object and sends the necessary instructions to the controller to generate the movements of the robot.

The system architecture is based on a client–server model in order to obtain scalability, better performance and modularity. This model helps to separate the tasks on different computers to reduce the computation time. In our work, the control system runs on the server whereas the vision system is a client residing on another computer (see Figure 2). The server must run and listen before the client tries to connect and, as soon as the connection is established, the client can send data to the server.

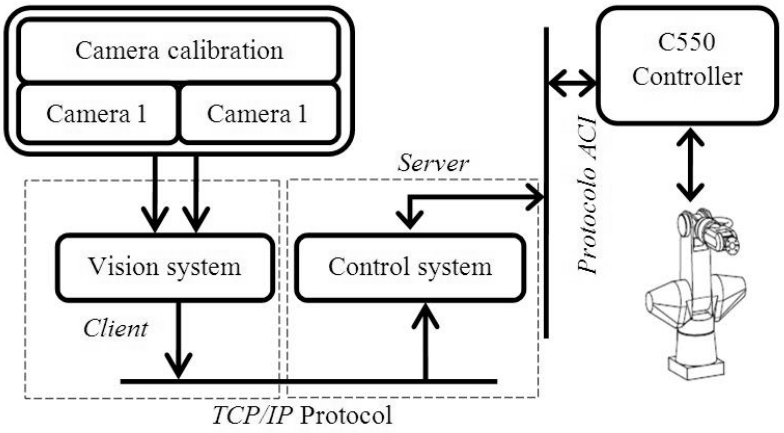


Figure 2. System configuration.

The communication between client and server is realized through an Ethernet network at 100 Mbps. To achieve the connection we use the TCP/IP (Transmission Control Protocol / Internet Protocol) protocol and sockets that keep an active line for the continuous transmission of data.

The C550 controller is connected to the computer through a serial connection at 38,400 bauds. The ACI (Advanced Communication Interface) communication protocol is used with the purpose of transmitting error-free data between the control system and the controller.

4. Camera Calibration

The main idea of calibrating a camera is to obtain the mathematical equations that relate coordinates W_i of points in the real world with coordinates w_i of their images [19], so it is necessary to calculate the intrinsic (depending on the camera) and the extrinsic (depending on the location of the camera in space) parameters *P*.

According to the pinhole camera model [16], camera matrix *P* is obtained, which is represented by Equation (1),

$$w_i = PW_i \tag{1}$$

where $w_i = (\rho x_i, \rho y_i, \rho)^{\mathrm{T}}$ is the homogeneous 3-vector that represents the points in the image, $W_i = (X_i, Y_i, Z_i, 1)^{\mathrm{T}}$ is the 4-vector that represents the points in the real world and *P* is the homogeneous *3 x 4* matrix in which are contained the intrinsic and extrinsic parameters of the camera. According to Equation (1), this may be written as:

$$\begin{bmatrix} \rho x_i \\ \rho y_i \\ \rho \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{2}$$

When multiplying the terms on the right hand side, we obtain:

$$\begin{bmatrix} \rho x_i \\ \rho y_i \\ \rho \end{bmatrix} = \begin{bmatrix} P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14} \\ P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24} \\ P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34} \end{bmatrix} \tag{3}$$

Both sides of the linear equation are divided by the last element:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \\ \dfrac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \\ 1 \end{bmatrix} \tag{4}$$

When equating components, we obtain two relations:

$$P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14} - x_iP_{31}X_i - x_iP_{32}Y_i - x_iP_{33}Z_i - x_iP_{34} = 0 \tag{5}$$

$$P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24} - y_iP_{31}X_i - y_iP_{32}Y_i - y_iP_{33}Z_i - y_iP_{34} = 0 \tag{6}$$

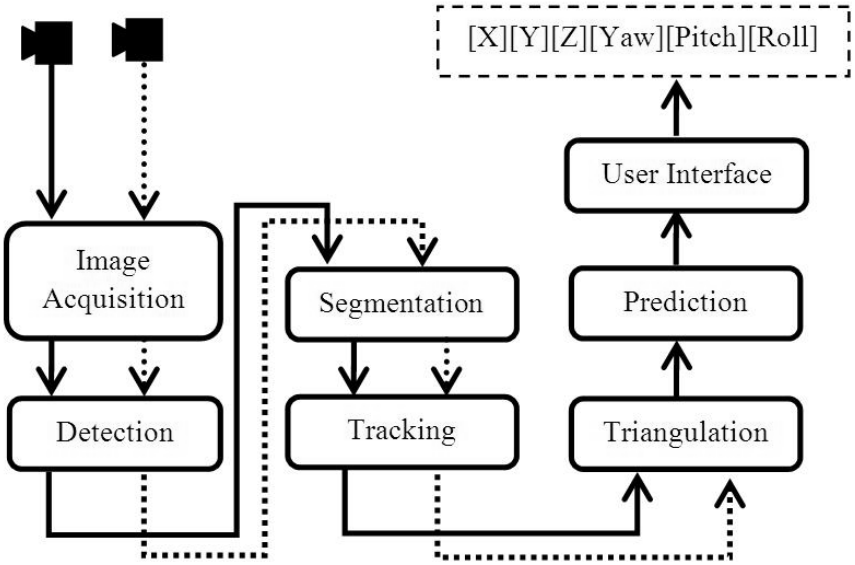


Figure 3. Detailed diagram of the vision system.

For stereo vision, two camera matrices *P* must be found. Equations (5) and (6) are obtained for each corresponding pair of points $\{w_i \leftrightarrow W_i\}$, for i = 1, 2..6, and for each one of two stereo images.

5. Vision system

The vision system is composed of the following modules: image acquisition, detection, segmentation, tracking, triangulation, prediction and user interface (see Figure 3).

The image acquisition module receives the video sequences coming from the cameras and divides them into frames for later analysis. No special illumination system is used to carry out the acquisition of the images. Natural daylight and ordinary fluorescent lamp lighting of the Artificial Intelligence Laboratory are used. With these two sources, it is possible to have a mostly diffuse illumination in a simple way, which presents a non directional character and produces very few shadows [3].

Color information of the object in movement (a red ball) is used as the main feature for its localization. As a result of a threshold process, binary images are obtained, where all the pixels that are the same shade of color as the object are identified.

When working in a real environment without controlled illumination, the apparition of phenomena like noise, shades or reflection that hinder the full localization of the object, is very common. These problems are eliminated in the segmentation module; objects of smaller size but with the same color as the object of interest are ignored.

In the tracking stage the center of mass or centroid of the group of pixels that pass the previous filters is determined in each one of the video frames. Once the coordinates of the object centroid are calculated in the images of the two cameras, the three dimensional position of the object is calculated by means of the triangulation module.

These coordinates are displayed in the user interface. The vision system is capable of updating the coordinates 28 times per second.

As will be explained in Section 6, the robot controller permits limited real time performance, so the vision system is operated at four frames per second. To compensate the delay that is produced by the response of the robot, we work out an estimate of the future coordinates of the object position in three dimensions, from a record of the past coordinates. This prediction, when fed back, allows generating a trajectory about sixteen sample times from the present position of the object (about four seconds).

5.1. Detection

This stage consists in identifying the object by its color and not by its movement or shape. Color is a frequently used feature in artificial vision to provide speed in the detection, segmentation and tracking process.

Scandaliaris and Villamizar [17] report that the RGB color model is inadequate for object detection, due to its sensitivity to shadows and reflections. Fuentes et al. [18] compare four color models, and conclude that the *normalized* RGB color model is a good tradeoff between computational cost and robustness for real time color detection. After capturing the image, the normalized values for R, G and B are

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)} \quad (7)$$

Next, we realize a thresholding process that allows filtering only the pixels that lie in a certain range in each one of the color bands. A threshold image $h(x,y)$ is defined by Equation (8), where $f(x,y)$ refers to the intensity of the original image in coordinates (x,y).

$$h(x,y) = \begin{cases} 0 & si \quad f(x,y) > T \\ 1 & si \quad f(x,y) \le T \end{cases} \quad (8)$$

Pixels marked with 1 correspond to the object, and those marked with 0 belong to the background.

The problem of computational cost of the color transformation is considerably reduced when applying the filtering of the image only to an area of interest. With this consideration, it is possible to work with a greater number of frames per second. These areas of interest are also known as search windows, which are usually located around the last known position of the object in each one of the images. Our search window is twice the apparent diameter of the object, which is at most 24% of the image width and 32% of the image height.

5.2. Segmentation

After the detection process, all pixels obtained are classified as part of the object. However, many of them are noise, reflection, shadows or other objects that are not of interest. An iterative algorithm for the labeling of related components is applied to carry out a refinement of the region of interest, which labels and groups pixels that are interconnected [20]. As a result of this process, different regions and their areas are obtained, and the largest region is taken to be the target.

5.3. Tracking

Visual tracking is the process of detecting in a repetitive way a feature, or set of features, in a sequence of images. There are several problems that complicate this process, from the noise originated in the video capture sensor to the variations of visibility that occur in the scene. In this work, a tracking technique based on the moments of the object, whose computational cost is relatively low, is used. The characteristic we track is the centroid of the region detected as the object.

Once the object is detected, the moments of order 0 and 1 are applied to calculate its centroid. Let $I(x,y)$ be the value of a pixel of the binarized image, the moment of order $(p+q)$ is defined by Equation (9), where *x* and *y* are the pixel coordinates, and *M* and *N* correspond to the size of the image.

$$M_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} x^p y^q I(x,y) \tag{9}$$

The central moments μ_{pq} of the image are expressed by means of Equation (10).

$$\mu_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} (x-x_c)^p (y-y_c)^q I(x,y) \tag{10}$$

Where (x_c, y_c) is the centroid of the object in the image, which is obtained by Equations (11).

$$x_c = \frac{M_{10}}{M_{00}}, y^c = \frac{M_{01}}{M_{00}} \tag{11}$$

According to [20] the central moments are easily calculated by means of Equations (12), (13), (14) and (15).

$$\mu_{00} = M_{00} \tag{12}$$
$$\mu_{20} = M_{20} - x_c M_{10} \tag{13}$$
$$\mu_{02} = M_{02} - y_c M_{01} \tag{14}$$
$$\mu_{11} = M_{11} - y_c M_{10} \tag{15}$$

The central moments are substituted in Equations (16) and (17) to calculate the major axis (l) and the minor axis (w).

$$l = \left[\frac{\mu_{20}+\mu_{02}+[(\mu_{20}-\mu_{02})^2+4\mu_{11}^2]^{1/2}}{\mu_{00}/2}\right]^{1/2} \tag{16}$$

$$w = \left[\frac{\mu_{20}+\mu_{02}-[(\mu_{20}-\mu_{02})^2+4\mu_{11}^2]^{1/2}}{\mu_{00}/2}\right]^{1/2} \tag{17}$$

The orientation angle of the major axis is calculated by Equation (18).

$$\theta = \frac{1}{2} a\tan 2\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{18}$$

Finally, to obtain the coordinates of the major and minor axes, the following formulas are applied:

$$l_1(x_1,y_1) = \left(x_c + \frac{l*\cos\theta}{2}, y_c + \frac{l*\sin\theta}{2}\right) \tag{19}$$

$$l_2(x_2,y_2) = \left(x_c - \frac{l*\cos\theta}{2}, y_c - \frac{l*\sin\theta}{2}\right) \tag{20}$$

$$w_1(x_3,y_3) = \left(x_c + \frac{w*\sin\theta}{2}, y_c - \frac{w*\cos\theta}{2}\right) \tag{21}$$

$$w_2(x_4,y_4) = \left(x_c - \frac{w*\sin\theta}{2}, y_c + \frac{w*\cos\theta}{2}\right) \tag{22}$$

With the result of this calculation, the algorithm draws both axes in the image and uses the final points *(l1, l2)* and *(w1, w2)* to define a bounding box.

5.4. Triangulation

Through artificial stereoscopic vision, it is possible to recover the geometric information of a point in space when its position is determined in two images taken by two calibrated cameras. This process requires calculating the intersection of two rays in space. This method is known as triangulation [16]. From several available methods described by Hartley and Sturm [21], we use the simplest and least accurate but fastest calculation, which was sufficient for our application.

The linear triangulation method is the most common one. Equation (1) holds $w = PW$. We write in homogeneous coordinates $w = \rho(x,y,1)^{\mathrm{T}}$, where (x,y) are the observed point coordinates in the image and ρ is a scale factor. Now, denoting by P_i^{T} , the i-th row of matrix P, this equation may be written as shown in Equation (23).

$$\rho x = P_1^T W, \rho y = P_2^T W, \rho = P_3^T W \tag{23}$$

Eliminating ρ using the third equation, we arrive at Equations (24) and (25).

$$xP_3^T W = P_1^T W \quad (24)$$

$$yP_3^T W = P_2^T W \quad (25)$$

We obtain a total of 4 linear equations relating the unknown real-world coordinates (X, Y, Z) of a point and the measured coordinates of its two images, which may be written in the form AX = B, as in Equation (26).

5.5. Prediction

In this work, the linear prediction method is used to predict the location of the centroid of the moving object in three dimensional space and not in images, based on the past value of the centroid coordinates. The basic idea of a linear predictor is that a sample in a given instant, $\hat{S}_n$, is approximated as a linear combination of the p previous samples [9], such that

$$\hat{S}_n = -\sum_{k=1}^{p} a_k S_{n-k} = -[a_1 S_{n-1} + a_2 S_{n-2} + \cdots + a_p S_{n-p}] \quad (27)$$

The prediction coefficients $a_k(a_1, a_2, \ldots, a_p)$ are calculated from the errors of forward and backward prediction [22]. The variance of the error is thus minimized in the prediction of the sample, as defined in (28)

$$\varepsilon = \sum_{n=p}^{N} [e_p^f(n)^2 + e_p^b(n)^2] \quad (28)$$

To obtain a_k, we used the Burg recursive algorithm to estimate the reflection coefficients r_k for k=1, 2, …, *p*, which are determined by expressions (29), (30), (31) and (32).

$$a_k(i) = \begin{Bmatrix} a_{k-1}(i) + r_k a_{k-1}(k-i) & for\, i = 1,2,\ldots,k-1 \\ r_k & for \quad i = k \end{Bmatrix} \quad (29)$$

$$r_k = \frac{-2\sum_{n=k}^{N-1} [e_{k-1}^f(n) e_{k-1}^b(n-1)]}{\sum_{n=k}^{N-1} [e_{k-1}^f(n)^2 + e_{k-1}^b(n-1)^2]} \quad (30)$$

$$e_k^f(n) = e_{k-1}^f(n) - r_k e_{k-1}^b(n-1) \qquad n = k+2, k+3, \ldots, N \quad (31)$$

$$e_k^b(n) = e_{k-1}^b(n-1) - r_k e_{k-1}^f(n) \qquad n = k+1, k+2, \ldots, N-1 \quad (32)$$

These are initialized with the values defined in Equations (33) and (34).

$$e_0^f(n) = x(n) \qquad n = 2,3,\ldots,N \quad (33)$$

$$e_0^b(n) = x(n) \qquad n = 1,2,\ldots,N-1 \quad (34)$$

6. Control system

The control system consists of a controller communication interface module and a grasping module. The grasping module controls the robot movements to grasp an object and the controller communication interface is responsible for providing an efficient communication between the control system and the C550 controller.

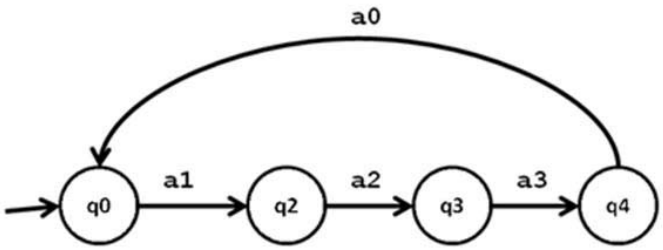
$$\begin{bmatrix} x_1 - P1_{14} \\ y_1 - P1_{24} \\ x_2 - P2_{14} \\ y_2 - P2_{24} \end{bmatrix} = \begin{bmatrix} (P1_{11} - x_1 P1_{31}) & (P1_{12} - x_1 P1_{32}) & (P1_{13} - x_1 P1_{33}) \\ (P1_{21} - y_1 P1_{31}) & (P1_{22} - y_1 P1_{32}) & (P1_{23} - y_1 P1_{33}) \\ (P2_{11} - x_2 P2_{31}) & (P2_{12} - x_2 P2_{32}) & (P2_{13} - x_2 P2_{33}) \\ (P2_{21} - y_2 P2_{31}) & (P2_{22} - y_2 P2_{32}) & (P2_{23} - y_2 P2_{33}) \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} \quad (26)$$

The standard communication interface with the C550 controller does not support real time interactions, so an interface was implemented with *Robwork* [23], a collection of C++ libraries developed in the robotics department of the Maersk Mc-Kinney Moller Institute at the University of Southern Denmark. However, the resulting system is painfully slow, largely due to the limited functionality of the C550 controller . The controller with this interface only permits one control interaction every 0.72 to 0.84 seconds, while the vision system operates at four frames per second.

We adopted the grasp manipulation scheme proposed by Kragic and Christensen [13]. It offers three basic steps that constitute a grip sequence: 1) approach or transport, 2) alignment and 3) grasping.

The approach or transport considers the movement of the robot arm toward the vicinity of the object. At the end of this step, the arm must be able to reach the object without the movement of its base. The alignment consists in aligning the gripper (end effector) according to the position of the object so that grasping may be carried out. The grasping is carried out once the position of the object is calculated. The grasping is made through a repetitive cycle of previously defined steps. The cycle is formulated by means of a finite state machine (see Figure 4). The basic states denoted by q_i represent the behavior of humans in grasping objects. Actions a_i are necessary to pass from one state to another.

We note that the robot response introduces a considerable delay which is overcome by means of the predictor. The position of the robot end effector is not directly verified by the vision system, and depends on the C550 controller and the calibration of the vision system with respect to the robot.



States	Actions
q0 = Hand opened	a0 = Open Hand
q2 = Grasping	a1 = Approaching object
q3 = Hand closed	a2 = Close hand
q4 = Object lifted	a3 = Lift object

Figure 4. Abstract representation of the grasping process.

7. Experimental Results

The most important components that made up the experimental platform of the system were the CRS T475 robot arm; the two computers that were required to install the system (conventional computers without special characteristics); the connection to the communication network, and the two Logitech QuickCam cameras fixed in a convergent configuration. The cameras were calibrated using a black and white checkerboard pattern, and mounted approximately 0.50 meters from each other with an angle of about 30 degrees from the line that unites the centers of the cameras.

In the experiments, a red ball was tracked by the vision system. The ball was moved around a circular trajectory plus a bidirectional displacement along de *Y* axis of the robot coordinate system. The speed of the circular motion was 1 rpm (revolution per minute) and the bidirectional displacement was carried out at 0.01 meters per second, approximately. The speed of the robot was fixed at 50% of its capacity. Figure 5 shows the real and predicted trajectory of the object projected in the *(X, Y).* plane, the ability of the vision system to calculate the position of the object in three dimensions from each pair of frames that conform the video sequence is visualized.

Figure 6 shows the behavior of the Burg predictor in the presence of occlusions in component Y. The crosses represent the visual measurements and the circles the predictor estimation. When the target is occluded, the values of the X, Y and Z coordinates returned by the tracking module are set to zero. While occlusion lasts, the predictor output is fed back as the correct target position, so it continues calculating the possible position of the object.

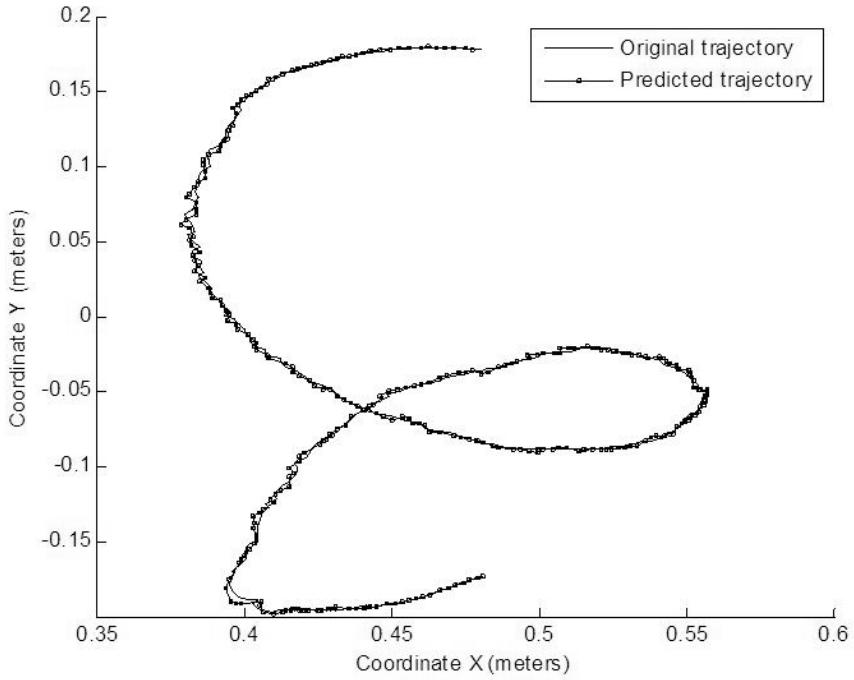


Figure 5. Points calculated by the vision system.

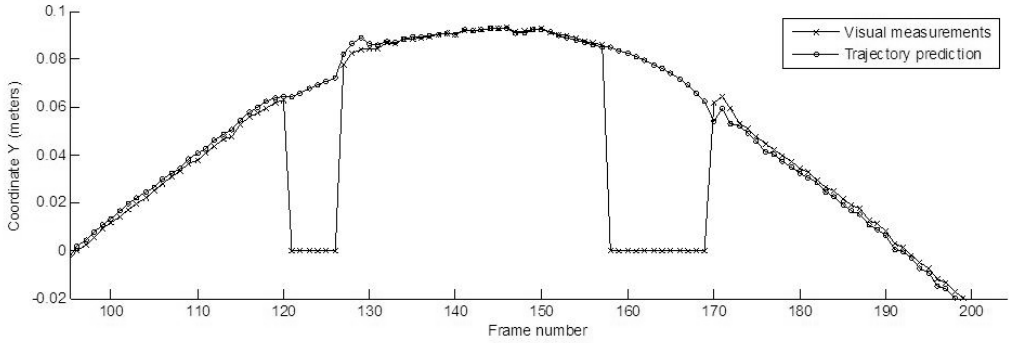


Figure 6. Predictor performance with total occlusions (component Y).

As shown in the image sequence of Figure 7, the tracking and grasping of the object is achieved efficiently. The left and right images of the first row show that the gripper of the robot keeps at a distance from the object while executing the tracking process. In the second row, the robot receives the grasping signal (left image) and opens the gripper (right image). Subsequently, the robot moves to the position of the centroid of the object (third row, left image) and finally closes the gripper (third row, right image), grasps the object and finally lifts it (not shown).

Figure 7. Successful grasping of a moving object.

Figure 8. Left and right camera images.

Figure 8 shows the left and right camera images of the moving object. In these figures, the tracking module marks the centroid of the object with a cross and displays its search window with a yellow box.

8. Conclusions

The analysis, design and implementation of a stereo vision system giving a robot manipulator the capability of tracking and grasping a rigid object moving in real time was reported.

This system was operated in ordinary laboratory conditions without special lighting. Based on the results obtained with the test cases, we show the system displays robustness, efficiency and low consumption of hardware resources to operate.

The tracking algorithm based on moments was effective to keep the object in the search window. Additionally, it was robust enough to work with partial or total occlusions for up to sixteen time samples (about 4 seconds).

Linear prediction through the method of Burg proved to be efficient in locally predicting the circular trajectory of the object in 3D space. Good results were also obtained when predicting the path produced by a circular motion plus a slow bidirectional displacement in the plane of the circle in spite of the fact that the predictor had been trained with a record of centroids obtained only by a circular movement. Using the linear predictor it was possible to compensate the delays caused by the response of the robot controller.

The great delay we have to cope with is not a common drawback on modern robots; however, it is present in many remote or telepresence applications, as Web access or unmanned orbital missions.

Although we used a simple grasping strategy, it provided good results for grasping a rigid object with a regular shape and complex movements. This strategy does not ensure optimum grip of objects because this would require an analysis of stable grasp points or having proximity sensors in the end effector. In this project, effectiveness depended entirely on the accuracy of the calibration and the prediction processes of the system.

References

[1] Fu, K., González, R., Lee, C., ROBOTICA: Control, Detección, Visión e Inteligencia, Editorial Mc Graw-Hill, (1988).

[2] Hutchinson, S., Hager, G., Corke, P., A Tutorial on Visual Servo Control, In IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, pp. 651-670, (1996).

[3] Corke, P., VISUAL CONTROL OF ROBOTS high-performance visual servoing, Editorial Research Studies Press Ltd., (1996).

[4] Allen, P., Timcenko, A., Yoshimi, B., et al., Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System, Proceedings of the IEEE Transactions on Robotics and Automation, Vol. 9, No. 2, (1993).

[5] Nomura, H., Naito, T., Integrated visual servoing system to grasp industrial parts moving on conveyer by controlling 6DOF arm, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 1768-1775, (2000).

[6] Sen, G., Messom, C., Demidenko, S., *et al*., Identification and prediction of a moving object using real-time global vision sensing, Proceedings of the 20th IEEE Instrumentation and Measurement Technology Conference, Volume 2, pp.1402-1406, (2003).

[7] Ge, L., Jie, Z., A Real-time Stereo Visual Servoing for Moving Object Grasping Based Parallel Algorithms, IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 2886-2891, (2007).

[8] Balkenius, C., Johansson, B., Event Prediction and Object Motion Estimation in the Development of Visual Attention, Proceedings of the Fifth International Workshop on Epigenetic, (2005).

[9] Makhoul, J., Linear prediction: A tutorial review, Proceedings of the IEEE, April, (1975).

[10] Yeoh, P., Abu-Bakar, S., Accurate real-time object tracking with linear prediction method, Proceedings of International Conference on Image Processing, Volume 3, (2003).

[11] Matas, J., Zimmermann, K., Learning efficient linear predictors for motion estimation, Proceedings of 5th Indian Conference on Computer Vision, Graphics and Image Processing, Springer-Verlag, Madurai, India, (2006).

[12] Ellis, L. Dowson, N., Linear Predictors for Fast Simultaneous Modeling and Tracking, Proceedings of IEEE 11th International Conference on Computer Vision, Brazil, (2007).

[13] Kragic, D., Christensen, H., A framework for visual servoing, Proceedings of the International Conference on Visual Systems, pp. 345-354, Austria, April, (2003).

[14] Kragic, D., Christensen, H., Survey on Visual Servoing for Manipulation, Technical report, Computational Vision and Active Perception Laboratory, (2002).

[15] Woods, A., Docherty, T., Koch, R., Image Distortions in Stereoscopic Video Systems, in Stereoscopic Displays and Applications IV, Proceedings of the SPIE, Volume 1915, San Jose, California, (1993).

[16] Hartley, R., Zisserman A., Multiple View Geometry in computer vision, Cambridge University Press, Second Edition, (2004).

[17] Scandaliaris, J., Villamizar, M., Robust Color Contour Object Detection Invariant to Shadows, 12th Iberoamerican Congress on Pattern Recognition (CIARP), Valparaiso, (2007).

[18] Fuentes, J., Ruíz, J., Rendón, J., Seguimiento y Asimiento de un Objeto en Movimiento por Medio de un Robot Manipulador y Visión Estéreo, Master thesis in Computer Science, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), México, (2009).

[19] Faugeras, O., Quang-Tuan, L., The Geometry of Multiple Images: The Laws that Govern the Formation of Multiple Images of a Scene and Some of their Applications, Editorial MIT Press, London, England, (2001).

[20] González R., Woods R., Tratamiento Digital de Imágenes, First Edition, Editorial Addison Wesley Iberoamericana, Wilmington, Delaware, U.S.A, (1996).

[21] Hartley, R., Sturm, P., Triangulation, Computer Vision and Image Understanding, Vol. 68, No.2, November, (1997).

[22] M. Lagrange, S. Marchand, M. Raspaud, Enhanced Partial Tracking Using Linear Prediction, Proceedings of the 6th Conference on Digital Audio Effects, London, (2003).

[23] RobWork, Yet Another Robotics Library. A framework for simulation and control of robotics with emphasis on industrial robotics and their applications. http://www.robwork.dk/.

Acknowledgments

This paper has been made possible thanks to the generous support from the following institutions which we are pleased to acknowledge: CONACYT (Consejo Nacional de Ciencia y Tecnología) and CENIDET (Centro Nacional de Investigación y Desarrollo Tecnológico).

Authors´ Biography

Jorge FUENTES-PACHECO

He received the B.S. degree from the Instituto Tecnológico de Chilpancingo, Mexico, in 2006. He obtained his M.Sc. degree in computer science from the Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Mexico, in 2009. He is currently a Ph.D. student at the CENIDET. His areas of professional interest are computer vision, robotics and artificial intelligence. In particular, he is interested in tracking, stereo vision, localization and 3D mapping for robot navigation in outdoor environments.

Jose RUIZ-ASCENCIO

J. Ruiz-Ascencio received the B. Sc. degree in physics from Universidad Nacional Autónoma de México (UNAM) in 1971, a M.Sc. degree from Stanford University in electrical engineering in 1973 and the D. Phil. degree from the University of Sussex in engineering and applied science in 1989. He was researcher at the Institute of Applied Mathematics, IIMAS-UNAM, full-time lecturer at the Universitat Autònoma de Barcelona, automation project leader for Allen-Bradley, researcher at the Instituto Tecnológico de Monterrey, and invited scholar at McGill University's Center for Intelligent Machines (2003-2004). He joined the Computer Science Department at the Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) in 1995 where he is a member of the Artificial Intelligence Group. His current interests are machine vision and intelligent control.

Juan Manuel RENDÓN-MANCHA

He received a PhD degree in computer science from the Université René Descartes (Paris 5) in 2002. He has a master's degree in artificial intelligence from the Université Pierre et Marie Curie (Paris 6) in 1998. He is an electronics engineer from the Instituto Tecnológico de La Laguna (1996). Since 2002, he has been an associate professor at the Facultad de Ciencias of the Universidad Autónoma del Estado de Morelos, Cuernavaca Mor., Mexico. His research interests include computer vision for mobile and manipulator robots and analysis in biomedical images.