Sensors **2010**, *10*, 7303-7322; doi:10.3390/s100807303

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Visual Control of Robots Using Range Images

Jorge Pomares *, Pablo Gil and Fernando Torres

Physics, Systems Engineering and Signal Theory Department, University of Alicante, PO Box 99, Alicante 03080, Spain; E-Mails: Pablo.Gil@ua.es (P.G.); Fernando.Torres@ua.es (F.T.)

* Author to whom correspondence should be addressed; E-Mail: jpomares@ua.es;
Tel.: +34-965-903-400, ext.: 2032; Fax: +34-965-909-750.

Received: 20 May 2010; in revised form: 23 July 2010 / Accepted: 28 July 2010 / Published: 4 August 2010

Abstract: In the last years, 3D-vision systems based on the time-of-flight (ToF) principle have gained more importance in order to obtain 3D information from the workspace. In this paper, an analysis of the use of 3D ToF cameras to guide a robot arm is performed. To do so, an adaptive method to simultaneous visual servo control and camera calibration is presented. Using this method a robot arm is guided by using range information obtained from a ToF camera. Furthermore, the self-calibration method obtains the adequate integration time to be used by the range camera in order to precisely determine the depth information.

Keywords: visual servoing; ToF cameras; self-calibration; robotics

1. Introduction

Nowadays, visual servoing is a well known approach to guide a robot using visual information. The two main types of visual servoing techniques are position-based and image-based [1]. The first one uses 3-D visually-derived information when making motion control decisions. The second one performs the task by using information obtained directly from the image. However, the interaction matrix employed in these visual servoing systems requires known different camera parameters and the depth of the image features.

A typical approach to determine the depth of a target is the use of multiple cameras. The most commonly applied configuration using more than one camera is stereo vision (SV). In this case, in order to be able to calculate the depth of a feature point by triangulation, the correspondence of this point in both cameras must be assured.

In this paper the use of 3D time-of-flight (ToF) cameras is proposed in order to obtain the required 3D information in visual servoing approaches. These cameras provide range images which give depth measurements of the visual features. In the last years 3D-vision systems based on the ToF principle have gained more importance compared to SV. Using a ToF camera, illumination and observation directions can be collinear, therefore, this technique does not produce incomplete range data due to shadow effects. Furthermore, SV systems have difficulties in estimating the 3D information of planes such as walls or roadways. They cannot find the corresponding physical point of the observed 3D-space in both camera systems. Hence the 3D information of that point cannot be calculated by applying the triangulation principle. Another standard technique to obtain 3D information is the use of laser scanners. The advantages of ToF cameras over laser scanners are the high frame rates and the compactness of the sensor. These aspects have motivated the use of a ToF camera to obtain the required 3D information to guide the robot.

Some previous works have been developed in order to guide a robot by visual servoing using ToF Cameras. Within these works, a visual servoing system using PSD (Position Sensitive Device) triangulation for PCB manufacturing is presented in [2]. In [3] a position-based visual servoing is described to perform the tracking of a moving sphere using a pan-tilt unit. In this last paper a ToF Camera manufactured by CSEM is used. A similar approach is described in [4] to determine object positions by means of an eye-to-hand camera system. Unlike these previous approaches, in this paper the range images are not used directly to estimate the 3D pose of the objects in the workspace. A new image-based visual servoing system which integrates range information in the interaction matrix is presented to perform the robot guidance. Another advantage of the proposed system over the previous ones is the possibility of performing the camera calibration during the task. To do so, the visual servoing system uses the range images not only to determine the depths of the features but also to adjust the ToF camera parameters during the task.

When a ToF camera is used, some aspects must be taken into consideration, such as large fluctuations in precision caused by external interfering factors (e.g., sunlight) and scene configurations (*i.e.*, distances, orientations and reflectivity). These influences produce systematic errors which must be processed. Specifically, the distance computed from the range images is very changing depending on the integration time parameter. This paper presents a method for the online adaptation of the integration time of ToF cameras. This online adaptation is necessary to capture the images in the best condition independently of the changes in distance (between camera and objects) caused by the movements of the camera when it is mounted on a robotic arm. Previous works have been developed for ToF camera calibration [5-7]. These works perform an estimation of the camera parameters and distance errors when static scenes are observed. In these researches, a fixed distance between the camera and the objects is considered. Therefore, these previous works cannot be applied in visual servoing tasks where the camera performs the tracking of a given trajectory. In this last case, the camera parameters such as the integration time must be modified in order to optimally observe the scene. To do this, several previous works adapt the camera parameters, such as the amplitude of the integration time, during the task. In [8] a CSEM-Swissrange camera is employed for the navigation of a mobile robot in an environment with different objects. This work automatically estimates the value of the integration time according to the intensity pattern obtained by the camera. However, this parameter is depends on illumination and reflectance conditions. To solve this problem, in [9] a PMD camera is also used for mobile robot navigation. This work proposes an algorithm based on the amplitude parameter. In contrast with [4], the range of working distance analyzed is between 0.25 m and 1 m for the application of visual servoing.

This paper is organized as follows: In Section 2, a visual servoing approach for guiding a robot by using an eye-in-hand ToF camera is presented. Section 3 describes the operation principle of the ToF cameras and the PMD camera employed. In Section 3, an offline camera calibration approach for computing the required integration time from an amplitude analysis is shown. In Section 5, an algorithm for updating the integration time during the visual servoing task is described. In Section 6, experimental results confirm the validity of the visual servoing system and the calibration method. The final Section presents the main conclusions.

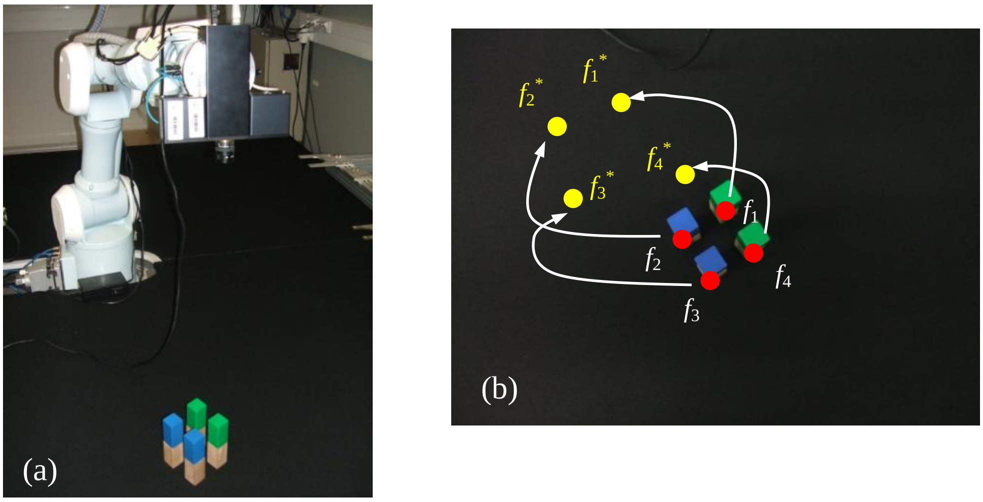
2. Visual Servoing Using Range Images

A visual servoing task can be described by an image function, $\mathbf{e}_t$, which must be regulated to 0:

$$\mathbf{e}_t = \mathbf{s} - \mathbf{s}^* \tag{1}$$

where $\mathbf{s} = (f_1, f_{2,} \ldots f_M)$ is a M × 1 vector containing M visual features observed at the current state ($f_i = (f_{ix}, f_{iy})$), while $\mathbf{s}^* = (f_1^*, f_2^*, \ldots f_M^*)$ denotes the visual features values at the desired state, *i.e.*, the image features observed at the desired robot location. In Figure 1(a) the eye-in-hand camera system is shown. A PMD19K camera is located at the end-effector of a 7 d.o.f Mitsubishi PA-10 robot that acquires grayscale images of 160 × 120. In Figure 1(b), an example of a visual servoing task is represented. This figure represents the initial and desired image features from the camera point of view.

Figure 1. (a) Eye-in-hand configuration. (b) Image acquired from the range camera point of view.



$\mathbf{L}_s$ represents the interaction matrix which relates the variations in the image with the variations in the camera pose [1]:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathrm{s}} \cdot \dot{\mathbf{r}} \qquad (2)$$

where $\dot{\mathbf{r}}$ represents the camera velocity.

By imposing an exponential decrease of $\mathbf{e}_{\mathrm{t}}$ ($\dot{\mathbf{e}}_{\mathrm{t}} = -\lambda_1 \mathbf{e}_{\mathrm{t}}$) it is possible to obtain the following control action for a classical image-based visual servoing:

$$\mathbf{v}_{\mathrm{c}} = -\lambda_1 \hat{\mathbf{L}}_{\mathrm{s}}^{+} \left(\mathbf{s} - \mathbf{s}^{*}\right) \qquad (3)$$

where $\lambda_1 > 0$ is the control gain, $\hat{\mathbf{L}}_{\mathrm{s}}^{+}$ is the pseudoinverse of an approximation of the interaction matrix and $\mathbf{v}_{\mathrm{C}}$ is the eye-in-hand camera velocity obtained from the control law in order to continuously reduce the error $\mathbf{e}_{\mathrm{t}}$. $\hat{\mathbf{L}}_{\mathrm{s}}^{+}$ is chosen as the Moore-Penrose pseudoinverse of $\hat{\mathbf{L}}_{\mathrm{s}}$ [1]. In order to completely define the control action, the value of the interaction matrix for the visual features extracted from the range images will be obtained in the following paragraphs.

First, the interaction matrix will be calculated when only one image feature (f_{x}, f_{y}) is extracted. The transformation between the range image $\mathbf{I}$(i,j) and 3D coordinates (relative to the camera position) is given by [10]:

$$\begin{aligned} x_{\mathrm{P}}^{\mathrm{C}} &= z_{\mathrm{P}}^{\mathrm{C}} \frac{f_{\mathrm{x}}^{'}}{\mathrm{s_x f}} \\ y_{\mathrm{P}}^{\mathrm{C}} &= z_{\mathrm{P}}^{\mathrm{C}} \frac{f_{\mathrm{y}}^{'}}{\mathrm{s_y f}} \\ z_{\mathrm{P}}^{\mathrm{C}} &= \mathbf{I}\left(f_{\mathrm{x}}, f_{\mathrm{y}}\right) \frac{\mathrm{f}}{\sqrt{\mathrm{f}^2 + \left(\frac{f_{\mathrm{x}}^{'}}{\mathrm{s_x}}\right)^2 + \left(\frac{f_{\mathrm{y}}^{'}}{\mathrm{s_y}}\right)^2}} \end{aligned} \qquad (4)$$

where f is the camera focal length, $\mathrm{s_x}$ and $\mathrm{s_y}$ are the pixel size in the x and y directions and $f_x^{'}, f_y^{'}$ are the normalized pixel coordinates, relative to the position ($\mathrm{u_0}$, $\mathrm{v_0}$) of the optical center on the sensor array $\left(f_x^{'} = f_x - \mathrm{u}_0, f_y^{'} = f_y - \mathrm{v}_0\right)$.

To obtain the interaction matrix, the intrinsic parameters $\xi = (\mathrm{f_u}, \mathrm{f_v}, \mathrm{u_0}, \mathrm{v_0})$ are considered, where $\mathrm{f_u} = \mathrm{f}\ \mathrm{s_x}$ and $\mathrm{f_v} = \mathrm{f}\ \mathrm{s_y}$. Therefore, considering these intrinsic parameters, Equation (4) is equal to:

$$\begin{aligned} x_{\mathrm{P}}^{\mathrm{C}} &= z_{\mathrm{P}}^{\mathrm{C}} \frac{f_x - \mathrm{u}_0}{\mathrm{f_u}} \\ y_{\mathrm{P}}^{\mathrm{C}} &= z_{\mathrm{P}}^{\mathrm{C}} \frac{f_y - v_0}{\mathrm{f_v}} \\ z_{\mathrm{P}}^{\mathrm{C}} &= \mathbf{I}\left(f_{\mathrm{x}}, f_{\mathrm{y}}\right) \frac{1}{\sqrt{1 + \left(\frac{f_{\mathrm{x}} - \mathrm{u}_0}{\mathrm{f_u}}\right)^2 + \left(\frac{f_{\mathrm{y}} - v_0}{\mathrm{f_v}}\right)^2}} \end{aligned} \qquad (5)$$

From (5) the coordinates of the image feature can be obtained as:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \mathrm{u}_0 \\ \mathrm{v}_0 \end{bmatrix} + \frac{1}{z_{\mathrm{P}}^{\mathrm{C}}} \begin{bmatrix} \mathrm{f}_\mathrm{u} & 0 \\ 0 & \mathrm{f}_\mathrm{v} \end{bmatrix} \begin{bmatrix} \mathrm{x}_\mathrm{P}^\mathrm{C} \\ \mathrm{y}_\mathrm{P}^\mathrm{C} \end{bmatrix} \tag{6}$$

The time derivative of the previous equation is:

$$\begin{bmatrix} \dot{f}_x \\ \dot{f}_y \end{bmatrix} = -\frac{\dot{z}_\mathrm{P}^\mathrm{C}}{\left(z_\mathrm{P}^\mathrm{C}\right)^2} \begin{bmatrix} \mathrm{f}_\mathrm{u} & 0 \\ 0 & \mathrm{f}_\mathrm{v} \end{bmatrix} \begin{bmatrix} \mathrm{x}_\mathrm{P}^\mathrm{C} \\ \mathrm{y}_\mathrm{P}^\mathrm{C} \end{bmatrix} + \frac{1}{z_\mathrm{P}^\mathrm{C}} \begin{bmatrix} \mathrm{f}_\mathrm{u} & 0 \\ 0 & \mathrm{f}_\mathrm{v} \end{bmatrix} \begin{bmatrix} \dot{\mathrm{x}}_\mathrm{P}^\mathrm{C} \\ \dot{\mathrm{y}}_\mathrm{P}^\mathrm{C} \end{bmatrix} \tag{7}$$

Considering the camera velocity $\dot{\mathrm{x}}_\mathrm{P}^\mathrm{C}, \dot{\mathrm{y}}_\mathrm{P}^\mathrm{C}, \dot{\mathrm{z}}_\mathrm{P}^\mathrm{C}$ divided in translational $\dot{\mathrm{x}}_\mathrm{t}^\mathrm{C}, \dot{\mathrm{y}}_\mathrm{t}^\mathrm{C}, \dot{\mathrm{z}}_\mathrm{t}^\mathrm{C}$ and rotational velocity $\dot{\alpha}^\mathrm{C}, \dot{\beta}^\mathrm{C}, \dot{\gamma}^\mathrm{C}$, the following expression can be obtained from Equation (7):

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{f}_\mathrm{x} \\ \dot{f}_\mathrm{y} \end{bmatrix} = -\frac{-\mathrm{x}_\mathrm{P}^\mathrm{C}\dot{\beta}^C + y_\mathrm{P}^\mathrm{C}\dot{\alpha}^C + \dot{z}_t^C}{\left(z_\mathrm{P}^\mathrm{C}\right)^2} \begin{bmatrix} \mathrm{f}_\mathrm{u} & 0 \\ 0 & \mathrm{f}_\mathrm{v} \end{bmatrix} \begin{bmatrix} \mathrm{x}_\mathrm{P}^\mathrm{C} \\ \mathrm{y}_\mathrm{P}^\mathrm{C} \end{bmatrix} + \frac{1}{z_\mathrm{P}^\mathrm{C}} \begin{bmatrix} \mathrm{f}_\mathrm{u} & 0 \\ 0 & \mathrm{f}_\mathrm{v} \end{bmatrix} \begin{bmatrix} -y_\mathrm{P}^\mathrm{C}\dot{\gamma}^C + z_\mathrm{P}^\mathrm{C}\dot{\beta}^C + \dot{x}_t^C \\ -z_\mathrm{P}^\mathrm{C}\dot{\alpha}^C + x_\mathrm{P}^\mathrm{C}\dot{\gamma}^C + \dot{\mathrm{y}}_\mathrm{t}^\mathrm{C} \end{bmatrix} \tag{8}$$

Developing the previous equation, an expression which relates the time derivative of the image features with the camera translational and rotational velocity can be obtained:

$$\dot{\mathbf{s}} = \underbrace{\begin{bmatrix} \frac{\mathrm{f}_\mathrm{u}}{z_\mathrm{P}^\mathrm{C}} & 0 & -\frac{(f_\mathrm{x}-\mathrm{u}_0)}{z_\mathrm{P}^\mathrm{C}} & -\frac{(f_\mathrm{x}-\mathrm{u}_0)(f_\mathrm{y}-\mathrm{v}_0)}{\mathrm{f}_\mathrm{v}} & \frac{(f_\mathrm{x}-\mathrm{u}_0)^2+\mathrm{f}_\mathrm{u}^2}{\mathrm{f}_\mathrm{u}} & -\frac{\mathrm{f}_\mathrm{u}(f_\mathrm{y}-\mathrm{v}_0)}{\mathrm{f}_\mathrm{v}} \\ 0 & \frac{\mathrm{f}}{z_\mathrm{P}^\mathrm{C}} & -\frac{(f_\mathrm{y}-\mathrm{v}_0)}{z_\mathrm{P}^\mathrm{C}} & -\frac{(f_\mathrm{y}-\mathrm{v}_0)^2-\mathrm{f}_\mathrm{v}^2}{\mathrm{f}_\mathrm{v}} & \frac{(f_\mathrm{x}-\mathrm{u}_0)(f_\mathrm{y}-\mathrm{v}_0)}{\mathrm{f}_\mathrm{u}} & \frac{\mathrm{f}_\mathrm{v}(f_\mathrm{x}-\mathrm{u}_0)}{\mathrm{f}_\mathrm{u}} \end{bmatrix}}_{\mathbf{L}_\mathrm{S}} \cdot \begin{bmatrix} \dot{x}_\mathrm{t}^\mathrm{C} \\ \dot{y}_\mathrm{t}^\mathrm{C} \\ \dot{z}_\mathrm{t}^\mathrm{C} \\ \dot{\alpha}^\mathrm{C} \\ \dot{\beta}^\mathrm{C} \\ \dot{\gamma}^\mathrm{C} \end{bmatrix} \tag{9}$$

where:

$$z_\mathrm{P}^\mathrm{C} = \mathbf{I}(f_\mathrm{x}, f_\mathrm{y}) \frac{1}{\sqrt{1 + \left(\frac{f_\mathrm{x}-\mathrm{u}_0}{\mathrm{f}_\mathrm{u}}\right)^2 + \left(\frac{f_\mathrm{y}-v_0}{\mathrm{f}_\mathrm{v}}\right)^2}} \tag{10}$$

The matrix obtained in Equation (9) is the interaction matrix, $\mathbf{L}_\mathrm{s}$, therefore, $\dot{\mathbf{s}} = \mathbf{L}_\mathrm{s} \cdot \dot{\mathbf{r}}$. The pseudo inverse of the interaction matrix derived in (9) is calculated for the control action in (3). In this last equation, an approximation of the interaction matrix is considered due the necessity of estimating the camera intrinsic parameters, ξ. If M visual features can be extracted from the image, the interaction matrix can be obtained as $\mathbf{L}_\mathrm{s}$= $[\mathbf{L}_{\mathrm{s}1}\ \mathbf{L}_{\mathrm{s}2} \ldots \mathbf{L}_{\mathrm{sM}}]^\mathrm{T}$, where $\mathbf{L}_{\mathrm{s}i}$ is the interaction matrix determined in (9) for only one feature.

Various previous works have studied the image-based visual servoing stability. In applications with commercial robots the complete dynamical robot model is not provided. In this cases, the system stability is deduced depending on kinematics properties [11-14]. Paper [1] describes that the local asymptotic stability can be ensured when the number of rows of the interaction matrix is greater than 6. However, we cannot ensure global asymptotic stability. As is indicated in [1], to ensure the local stability, the desired visual features must be closed the current ones. Furthermore, $\hat{\mathbf{L}}_\mathrm{s}^+$ and $\mathbf{L}_\mathrm{s}^+$ must be equal or very similar. To do so, the camera depth and intrinsic parameters must be correctly computed. The following algorithm [15] has been used in order to estimate the camera intrinsic parameters. In addition, the accurate determination of the camera depth is one of the main problems. It will be solved in the following sections.

3. Analysis of the Distance Measurement Computed with the ToF Camera

In this section, a behaviour analysis of ToF cameras is provided. This analysis helps to define the methods to improve the depth measurement which will be used in the visual servoing system. A PMD19K camera has been used in this analysis. The PMD19K camera contains a Photo Mixer Device (PMD) array with a size of 160 × 120 pixels. This technology is based on CMOS technology and a time-of-flight (ToF) principle.

There are other similar cameras based on the same principle and with CMOS technology such as the CamCube 2 or 3 of PMD-Technologies and the SR2, SR3000 or SR4000 of CSEM-Technologies. The specifications and a comparison of the behaviour of these cameras is available in [8] and [16], respectively. PMD19K works with a wavelength of near-infrared (NIR) light of 870 nm and it can capture up to 15 fps with a depth resolution of 6 mm. Furthermore, in the experiments here presented, the camera is connected by Ethernet and it is programmable by SDK for Windows, although it can be connected by Firewire interface and programmable for Linux, too. The ToF camera technology is based on the principle of modulation interferometry [6,16]. The scene is illuminated with NIR light (PMD19K module with a default frequency of ω = 20 Mhz) and this light is reflected by the objects in the scene. The difference between both signals, emitted and reflected, causes a phase delay which is detected for each pixel and used to estimate the distance value. Thus, the ToF camera provides $2\frac{1}{2}$ D depth information of dynamic or static scenes irrespective of the object’s features such as: intensity, depth and amplitude data simultaneously for each pixel of each image captured. The intensity represents the grayscale information, the depth is the distance value calculated within the camera and the amplitude is the signal strength of the reflected signal (quality of depth measures). Then, given the speed light, c, the frequency modulation, ω, the correlation between signals for four internal phase delays, $r_0(0^\circ)$, $r_1(90^\circ)$, $r_2(180^\circ)$, $r_3(270^\circ)$, the camera compute the phase delay, ϕ , the amplitude, a, and the distance between sensor and the target, z, as follows:

$$\varphi = \arctan\left(\frac{r_1 - r_3}{r_0 - r_2}\right) \tag{11}$$

$$a = \frac{\sqrt{(r_1 - r_3)^2 + (r_0 - r_2)^2}}{2} \tag{12}$$

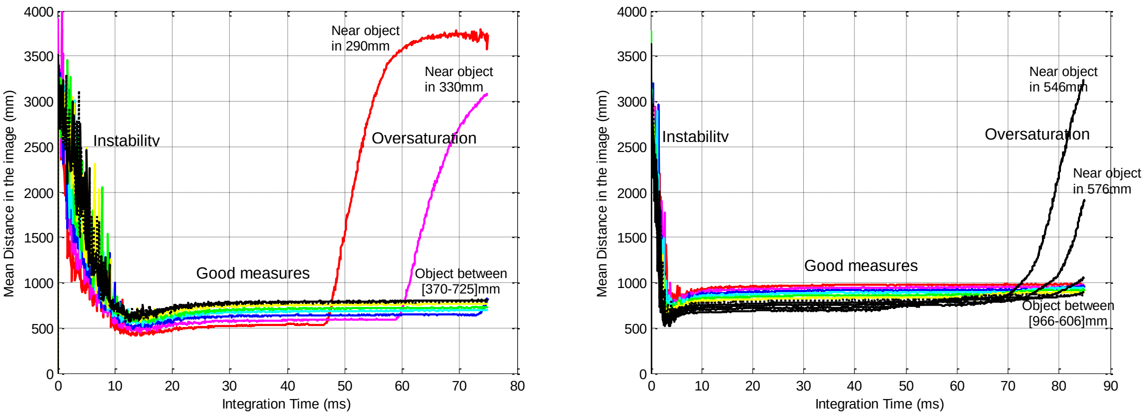
$$z_P^C = \frac{c\varphi}{4\pi\omega} \tag{13}$$

This type of cameras has some disadvantages [17]: they are sensitive to background light and interferences and they cause oversaturation and underexposure pixels. The PMD camera has two adjustable parameters to attenuate these errors in the pixels: the modulation frequency and the integration time. To do not change the original calibration determined by the manufacturer, only the behaviour of integration time has been studied to be adjusted. The integration time is defined as the exposure time or the effective length of time a camera’s shutter is open. This is time is needed so that the light reachs the image sensor suitably.

In a visual servoing system with eye-in-hand configuration (Figure 1) the camera is mounted at the end-effector of a robotic arm. Therefore, when the robot is moved, the distance between sensor and target, z_P^C, changes and the integration time, τ, has to be on-line adjusted to minimize the error in the computed depth. Whenever this parameter is suitably computed, the image range can be acquired in better conditions and so the features extraction process in the image can be improved looking for reaching the best features without modified the light environment or the object surfaces in the scene.

Figure 2 shows the stability of the distance measurements obtained from the range images with regards to the integration time. May *et al.* [9] show this dependency in a Swisrange SR-2 camera for the navigation of a mobile robot. The same is studied by Wiedemman *et al.* [8] to build maps with a mobile robot and by Gil *et al.* [17] to guide a robotic arm by using an eye-in-hand configuration for visual servoing (Figure 1). In this last work, a PMD19K camera was used.

Figure 2. Evolution of the mean distance of the range image for two different scenes: (a) An object and the camera moved between 0.5 m and 1 m. (b) Four objects and the camera moved between 0.3 m and 0.8 m.



(a) (b)

In previous works, some experiments were done in order to observe the evolution of the distance measured by the camera when the integration time changed. In those experiments from 750 images (an integration time offset of 100 ms between each image), a relationship between mean distance value, z_P^C, and integration time, τ, in microseconds is shown when the robot (Figure 1) is moved and the distance between sensor and target changes. As Figure 2 shows, when integration time is small, the distance computed is unstable and nontrustworthy. In the same way, when the integration time is high an oversaturation phenomenon sometimes appears in the signal which determines the distance curves. Normally, this phenomenon only appears when the distance measured between scene and camera is below a fixed nominal distance or distance threshold, as it is explained in [17]. In Figure 2(a), oversaturation appears when the integration time is greater than 45 ms. However, in Figure 2(b), the oversaturation only occurs when the integration time is greater than 70 ms. Therefore, the nearer the target is, the smaller the threshold of integration time must be. Thus, the farther the target is, the more precise the distance computed is. In addition, something similar happens with the intensity as it is explained in [9], although it is more sensitive to the background light and interferences [8,12]. Consequently, in the calibration process, the flat zone of the curve (Figure 2) has to be computed in order to use a ToF camera such as PMD19K for visual servoing. This zone determines the minimum and maximum integration times allowed to avoid the oversaturation and the instability problems. In this paper, these values have been fixed using the calibration method presented in [17], where the histogram which represents the frequency distributions of the amplitude measurements of PMD19k are adjusted by means of probability density functions (PDF) using Kolmogorov-Smirnov and Anderson-Darling methods.

4. Camera Calibration: Computing Integration Time from an Amplitude Analysis.

As regards the amplitude measurements, the curve which shows the evolution of the mean amplitude can be computed from a set of images acquired using a nominal fixed distance (the same as the mean distance that was computed in Figure 2). The analysis of the mean amplitude curve determines the thresholds of time integration, [τ_{min}, τ_{max}] which are needed in order to guarantee the precise computation of the distance measurements (Figure 3). The amplitude parameter, a, of a ToF camera defines the quality of the range images computed using a specific integration time. The minimum threshold, τ_{min}, is computed as the minimum integration time needed to compute the image depth in the desired camera location. It is determined as the time value where a least squares line fitting the mean amplitude curve crosses the zero axis (Figure 3). The maximum threshold, τ_{max}, is computed as the maximum integration time needed to compute the image depth in the initial camera location. These limits (Figure 3) are computed depending on the distance between target and camera by means of an offline process, as follows:

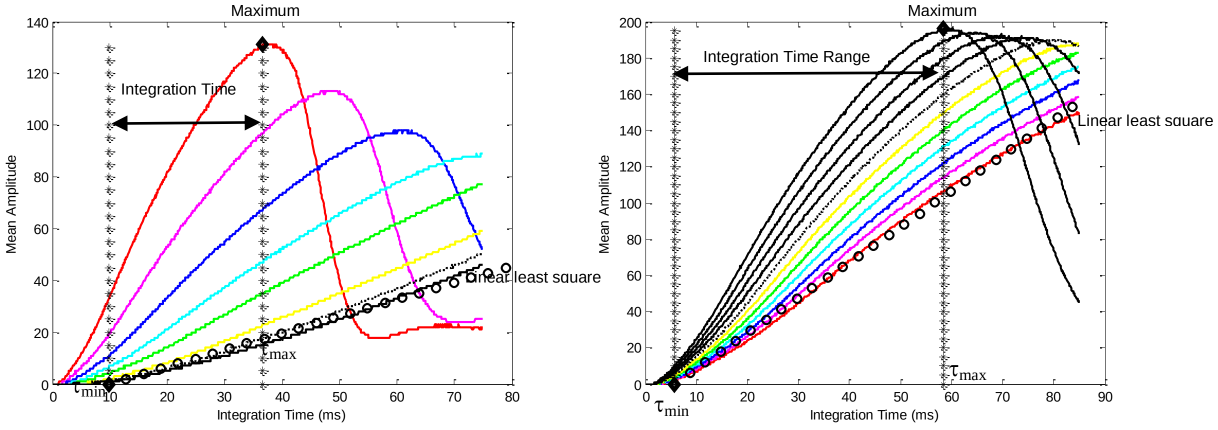
Pose the Robot in the initial pose and capture an image, I_τ, for some integration time, $\tau \in [0{,}85ms]$
At each iteration:

1. Compute mean amplitude: a_m
2. Estimate the frequency histogram for a_m and fit it by means of K-S and A-D Tests in order to classify the scene according to look-up-table as near or far target
    a. τ_{min} is computed from zero crossing determinated by the fitting of the curve which represents the image at the maximum distance ($\min\{\tau\}$ to capture the image at maximum working distance) (see Figure 3)
    b. τ_{max} is computed as the suitable integration time for obtaining a desired mean amplitude, a_d, such as:

If (near) then $a_d = \max\{a_m\}$ else $a_d = \mathrm{upper\_quartile}\{a_m\}$

The amplitude analysis of Figure 3 shows a group of curves (a curve for each camera location). The curves show how the linearisation level (part of flat slope) determines the degree of oversaturation. Thus, the amplitude curves grow quickly until they reach an absolute maximum value when the camera is near the target and the curves are more linear when the camera is moved away from the target.

Figure 3. Evolution of mean amplitude, a_m, for the tests of Figure 2.



(a) (b)

Once, the integration time values for final and initial camera positions have been computed, some intermediate integration time, τ_k, Figure 3(a) are computed for the robot trajectory. To do this, empirical tests have been done with the following algorithm:

1. Fix the integration time as $\tau_0 = \tau_{max}$ for image I_0
2. Compute the deviation error $e_a = a_d - (a_m)_0$ where $a_d = \max\{a_m\}$ according to a desired minimum distance.
3. Update integration time following the control law $\tau_k = \tau_{k-1}(1 + K \cdot e_a)$ where K is a proportional constant and it is adjusted depending on the robot velocity.

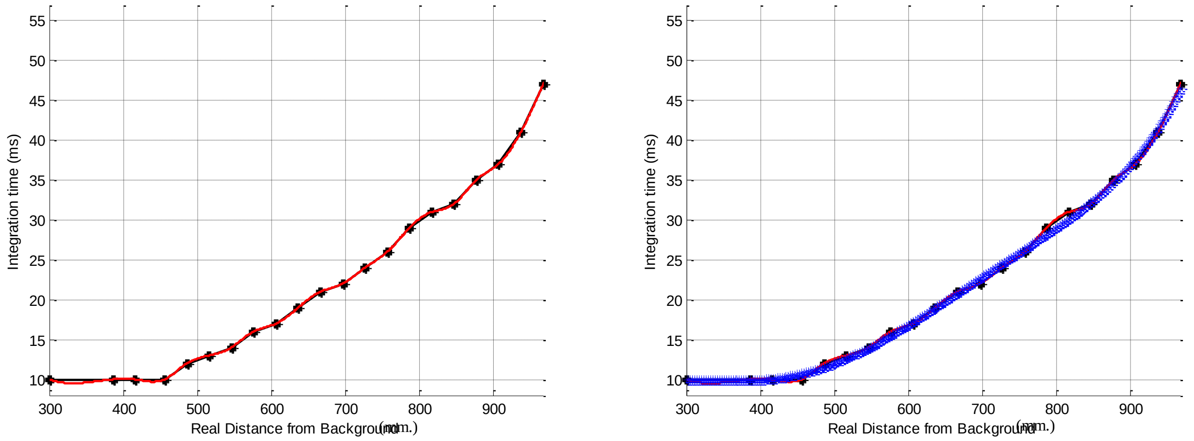
This way, some intermediate integration time values, $\tau_k \in [\tau_{min}, \tau_{max}]$, have been estimated for different distances between the final and the initial positions. Therefore, the proper computation of $\frac{\partial \tau}{\partial z_P^C}$ is done using a polynomial interpolation which fits these intermediate positions (Figure 4). In general, polynomial interpolation may not fit precisely at the end points. But this is not a problem because they are fixed with the time integration needed for the desired and the initial camera positions. Considering, τ_{min} and τ_{max} as the values 10 ms and 46.4 ms (upper quartile of the maximum value shown in Figure 3(b), 57.4 ms.) respectively and some intermediate time, τ_k, all computed, according the previous calibration method, $\frac{\partial \tau}{\partial z_P^C}$ is given by:

$$\frac{\partial \tau}{\partial z_P^C} = 2.8825\, z_\mu^4 + 4.5556\, z_\mu^3 - 4.581 z_\mu^2 + 0.4968\, z_\mu + 11.8853 \tag{14}$$

where:

$$z_\mu = \frac{z_P^C - 662}{193} \tag{15}$$

Figure 4. Polynomial interpolation applied to compute $\frac{\partial \tau}{\partial z_{\mathrm{P}}^{\mathrm{C}}}$ for distances between 0.3 and 1 m, for the tests of Figure 2.



(a) (b)

5. Algorithm for Updating the Camera Integration Time During the Task

From the previous analysis, a method to automatically update the integration time is presented in this section in order to be applied during visual servoing tasks.

Considering ${}^{c}\mathbf{M}_{o}$ the extrinsic parameters (pose of the object frame with respect to the camera frame), an object point can be expressed in the camera coordinate frame as:

$$\mathbf{P}_{\mathrm{P}}^{\mathrm{C}}\left(x_{\mathrm{P}}^{\mathrm{C}}, y_{\mathrm{P}}^{\mathrm{C}}, z_{\mathrm{P}}^{\mathrm{C}}\right) = \mathbf{M}_{\mathrm{O}}^{\mathrm{C}} \mathbf{P}_{\mathrm{P}}^{\mathrm{O}} \tag{16}$$

Considering a pin-hole camera projection model, the point $\mathbf{P}_{\mathrm{P}}^{\mathrm{C}}$ with 3D coordinates relative to the camera reference frame is projected onto the image plane at the point **p** of 2D coordinates. This point is computed from the focal length (distance between retinal plane and optical center of camera) as:

$$\mathbf{p} = (x, y)^{\mathrm{T}} = \left(\mathrm{f}\,\frac{x_{\mathrm{P}}^{\mathrm{C}}}{z_{\mathrm{P}}^{\mathrm{C}}}, \mathrm{f}\,\frac{y_{\mathrm{P}}^{\mathrm{C}}}{z_{\mathrm{P}}^{\mathrm{C}}} \right)^{\mathrm{T}} \tag{17}$$

Finally, the units of (17) specified in terms of metric units (e.g., mm.) are scaled and transformed in coordinates in pixels relative to the image reference frame, as:

$$\mathbf{s} = \left(f_{\mathrm{x}}, f_{\mathrm{y}}\right) = \left(\mathrm{u}_0 + \mathrm{f}_{\mathrm{u}} x, \mathrm{v}_0 + \mathrm{f}_{\mathrm{v}} y\right) \tag{18}$$

where $\xi = (\mathrm{f}_{\mathrm{u}}, \mathrm{f}_{\mathrm{v}}, \mathrm{u}_0, \mathrm{v}_0)$ are the camera intrinsic parameters.

The intrinsic parameters describe properties of the camera used, such as the position of the optical center (u_0, v_0), the size of the pixel and the focal length defined by (f_{u}, f_{v}). They are computed from a calibration process based on [15]

During a visual servoing task, the camera extrinsic parameters are not known, and ${}^{c}\mathbf{M}_{o}$ is considered as an estimation of the real camera pose. In order to determine this pose, we must minimize progressively the error between the observed data, $\mathbf{s}_{o}$, and the position of the same features computed by back-propagation employing the current extrinsic parameters, **s** (16)–(18). Therefore an error function which must be progressively reduced is defined as:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}_o \tag{19}$$

The time derivative of **e** will be:

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} - \dot{\mathbf{s}}_o = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathrm{t}} = \mathbf{L}_s \frac{\partial \mathbf{r}}{\partial \mathrm{t}} \tag{20}$$

To make **e** decrease exponentially to 0, $\dot{\mathbf{e}} = -\lambda_2 \mathbf{e}$, we obtain the following control action:

$$\frac{\partial \mathbf{r}}{\partial \mathrm{t}} = -\lambda_2 \mathbf{L}_s^+ \mathbf{e} \tag{21}$$

where λ_2 is a positive control gain and $\mathbf{L}_s^+$ is the pseudoinverse of the interaction matrix (9). Once the error is annulled the extrinsic parameters will be obtained. This approach is used by the virtual visual servoing systems to compute the camera locations. More details about the convergence, robustness and system stability can be seen in [11,12].

Consequently, two estimations are obtained for the depth of a given image feature: one depth (z_1) from the previous estimated extrinsic parameters and another depth ($z_2 = z_P^C$) from (10). This last depth is calculated from the range image and, therefore, can be updated by modifying the camera integration time. The adequate integration time will be obtained when z_1 and z_2 are equal. Therefore, a new control law is applied in order to update the integration time, τ, by minimizing the error between z_1 and z_2:

$$\frac{\partial \tau}{\partial \mathrm{t}} = -\lambda_3 \frac{\partial \tau}{\partial z_p^C} (z_2 - z_1) \tag{22}$$

where $\lambda_3 > 0$.

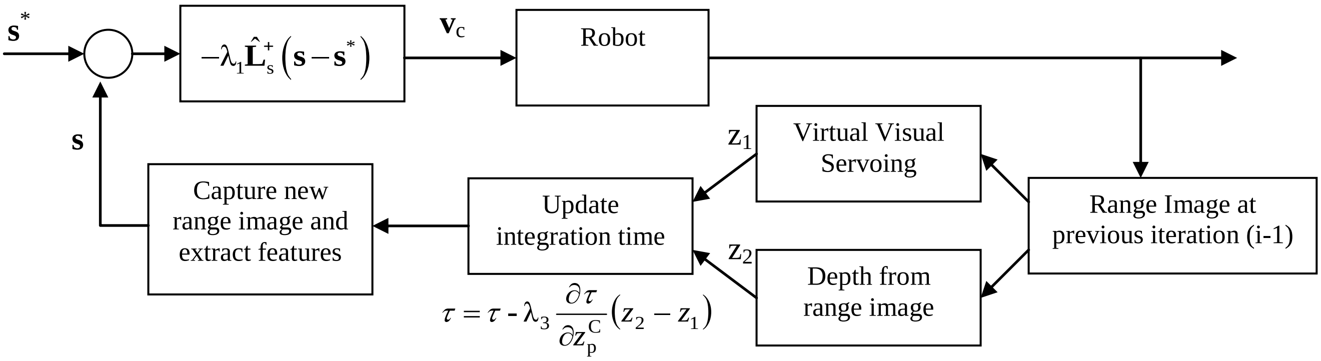
The algorithm for updating the camera integration time is summarized in the following lines:
First perform the offline camera calibration to determine the initial integration time and $\frac{\partial \tau}{\partial z_p^C}$ (see Section 4).

At each iteration of the visual servoing task:

1. Apply the control action to the robot: $\mathbf{v}_c = -\lambda_1 \hat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*)$
2. Estimate the extrinsic parameters using virtual visual servoing.
3. Determine the depth, z_1, from the previous extrinsic parameters and z_2 from the range image (10).
4. Update the integration time by applying $\frac{\partial \tau}{\partial \mathrm{t}} = -\lambda_3 \frac{\partial \tau}{\partial z_p^C} (z_2 - z_1)$

In order to describe more clearly the interactions among all the subsystems that compose the proposed visual servoing system, a block diagram is represented in Figure 5. In this block diagram (Figure 5) it is possible to observe that in the feedback of the visual servoing system a complete convergence of virtual visual servoing is performed in order to determine the extrinsic parameters. Moreover, the convergence and stability aspects when virtual visual servoing techniques are used as feedback of a visual servoing system are discussed in [18].

Figure 5. Block diagram of the visual servoing system.


$\mathbf{s}^*$
$-\lambda_1\hat{\mathbf{L}}_s^+(\mathbf{s}-\mathbf{s}^*)$
$\mathbf{v}_C$
Robot
s
z_1
Virtual Visual Servoing
Capture new range image and extract features
Update integration time
Range Image at previous iteration (i-1)
z_2
Depth from range image
$\tau = \tau - \lambda_3 \frac{\partial \tau}{\partial z_p^C}(z_2 - z_1)$

6. Results

The target used for the experiments can be seen in Figure 1. This target is composed of four objects on a black table as background to ensure a low reflectivity at the borders. The PMD19k is mounted at the end-effector of a Mitsubishi PA10 with 7 d.o.f. In addition, the ambient light (exterior light source) was controlled with a power regulator for this work in indoor environments. Thereby, special care was taken to avoid the interference with the NIR of the camera.

6.1. Trajectory 1

The real distance between camera and target (background and objects) for this first experiment was 600 < z_P^C < 966 mm. The initial and final camera locations were $\mathbf{P}_{Pi}^C = (0,0,966)\text{mm}$ and $\mathbf{P}_{Pf}^C = (-100,-200,600)\text{mm}$ respectively. The features are computed as the centroid of the four objects represented in the range image acquired by the PMD19k (Figure 1). The pixel coordinates of these centroids are $\mathbf{p}_1 = (7,23)^T$, $\mathbf{p}_2 = (27,12)^T$, $\mathbf{p}_3 = (17,41)^T$ and $\mathbf{p}_4 = (37,29)^T$ for the initial robot pose and $\mathbf{p}_1 = (85,40)^T$, $\mathbf{p}_2 = (115,24)^T$, $\mathbf{p}_3 = (103,71)^T$ and $\mathbf{p}_4 = (134,52)^T$ for the final pose. Figure 6 depicts the initial and final positions of the visual features and the eye-in-hand camera.

In Figure 7, the measured depth data from a range image is shown for three different camera locations. Only a range image was plotted but from three different camera location (offset $\Delta\mathbf{P} = (\Delta X, \Delta Y, \Delta Z)\text{mm}$ between locations) with the same time integration value, 53 ms. This plot shows distinct systematic errors when the integration time is not updated or it is chosen inadequately. However, these errors can be easily corrected by applying the method presented in Section 5. Thus, the combination of the calibration method for estimating the integration time in the initial position [17] and the method to update the integration time presented in sections 4 and 5 significantly improves the quality of the measured depth data.

Figure 6. (a) Initial position of the image features and the eye-in-hand camera. (b) Final position of the image features and the eye-in-hand camera. (Trajectory 1).

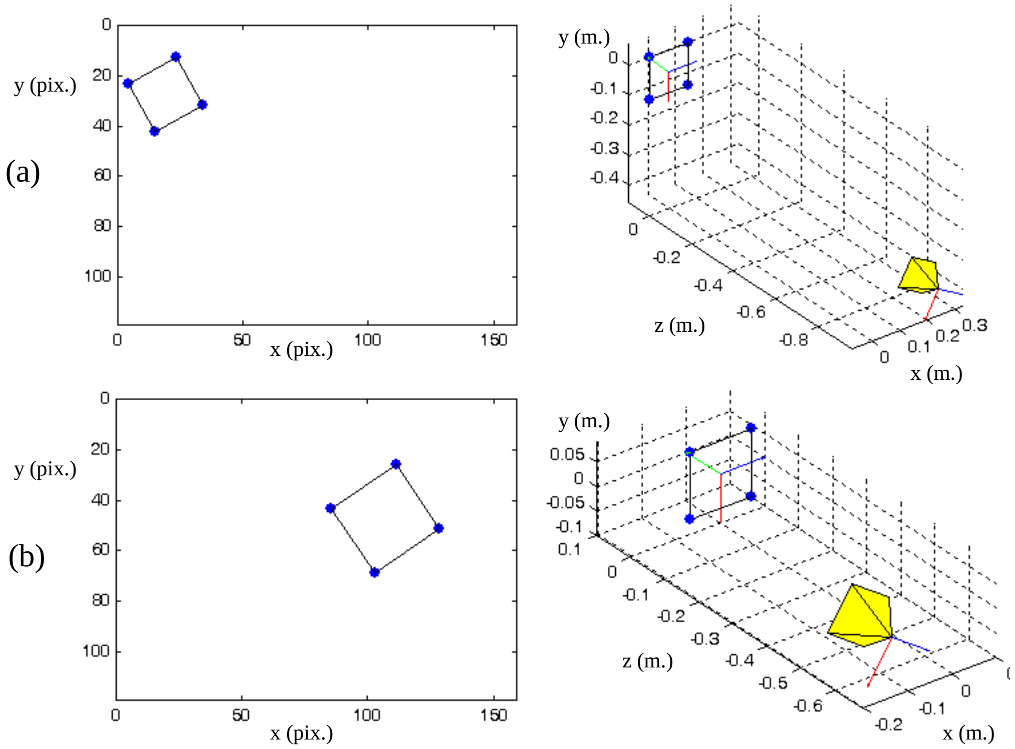
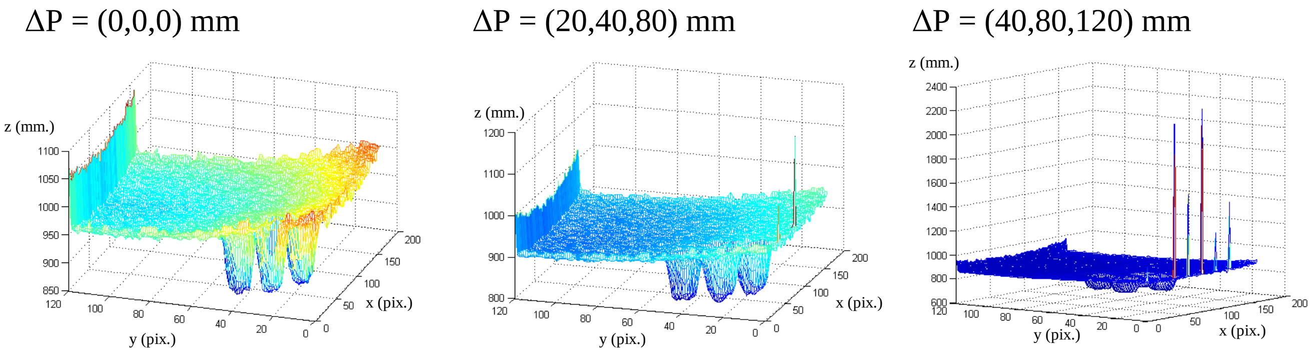


Figure 7. Range Image computed for the integration time of 53 ms.

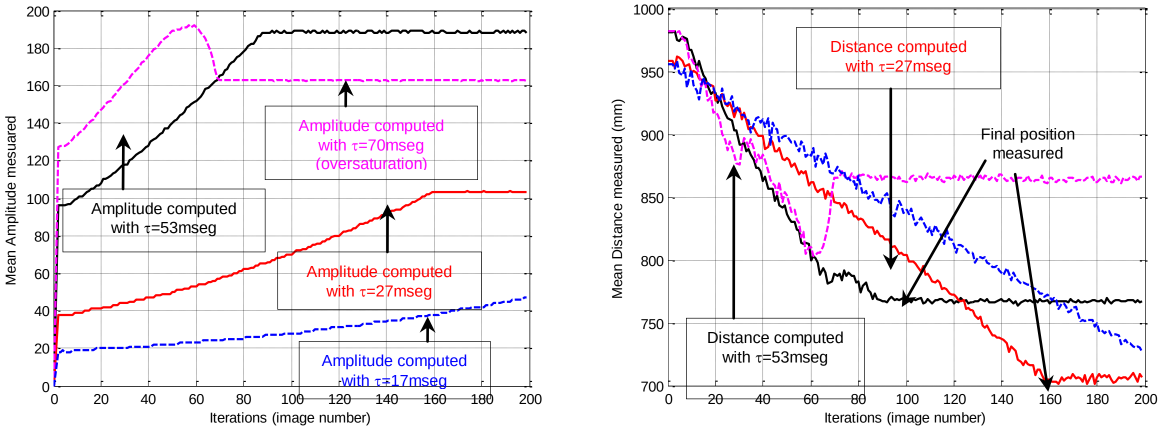

ΔP = (0,0,0) mm

ΔP = (20,40,80) mm

ΔP = (40,80,120) mm

Furthermore, Figure 8 shows how the depth and amplitude measured by the PMD19k change when the integration time is not updated to adapt it according to the distance between camera and target when the robot is moving. The PMD19K has been configured with some different integration times (17, 27, 53 and 70 ms). For example, 53 ms and 27 ms are near the good integration times for the initial and final camera locations, respectively. The experimental results show that whenever an integration time is greater than the optimal value (such as 70 ms), the amplitude values show instability after the maximum amplitude is reached (Figure 8(a)). Furthermore, if the used integration time is smaller than the optimal value (such as 17 ms), so many iterations are needed until the distance is computed correctly (Figure 8(b)). However, the time 27 ms compute a depth for the final position close to the final camera location.

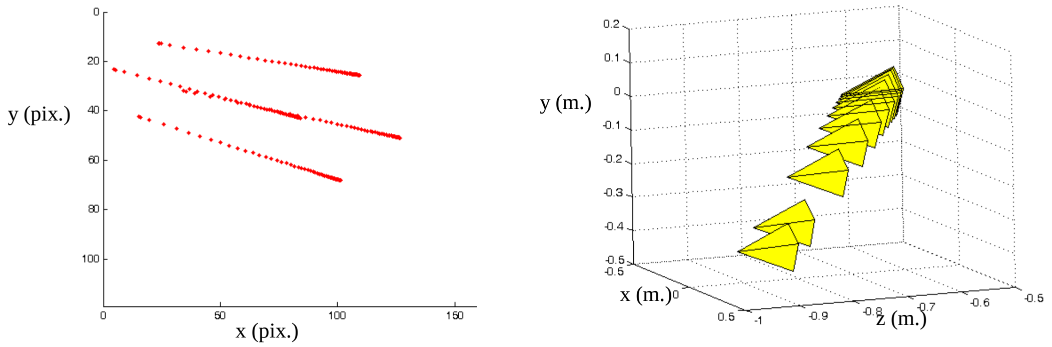
Figure 8. (a) Evolution of the measured amplitude when the integration time is not updated. (b) Evolution of the depth parameter when the integration time is not update.



(a) (b)

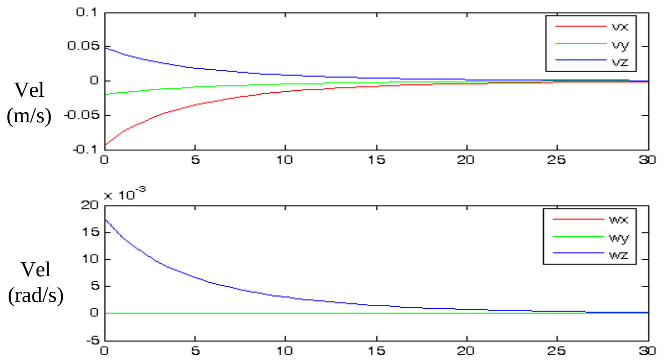
Applying the algorithm described in Section 5 from the initial and desired image features location, the image trajectory presented in Figure 9(a) is obtained. In this figure, it is possible to observe that the image features follow a straight line between the initial and the final locations. Furthermore, in Figure 9(b), the camera poses during the visual servoing task are represented. It is possible to observe that the visual servoing task is correctly performed. Therefore, we can conclude that a correct behaviour is obtained in the image and in the 3D space. In Figure 10 the velocities of the robot's end-effector applied during the visual servoing task are represented.

Figure 9. Trajectory during the visual servoing task. (a) Trajectory of the image features. (b) Trajectory of the eye-in-hand camera. Experiment 1.



(a) (b)

Figure 10. Velocities during the visual servoing task. Experiment 1.



In order to perform the correct tracking, the integration time is updated at each iteration of the visual servoing task using the algorithm described in Section 5. Figure 11 shows the value of the integration time considered at each iteration. Finally, considering these values of the integration time, the new range images obtained at ΔP = (0,0,0) mm, ΔP = (20,40,80) mm and ΔP = (40,80,120) mm are represented in Figure 12. Comparing these figures with the ones obtained at Figure 7, it is possible to observe that the update process of the integration time based on the proposed algorithm eliminates the the previous errors.

Figure 11. Integration time values at each iteration of the visual servoing task. Trajectory 1.

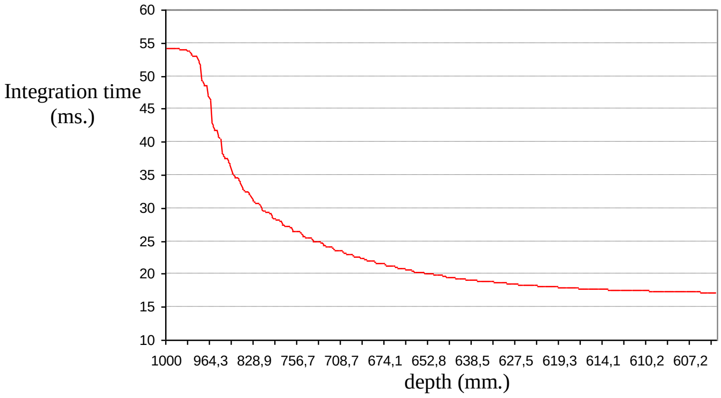
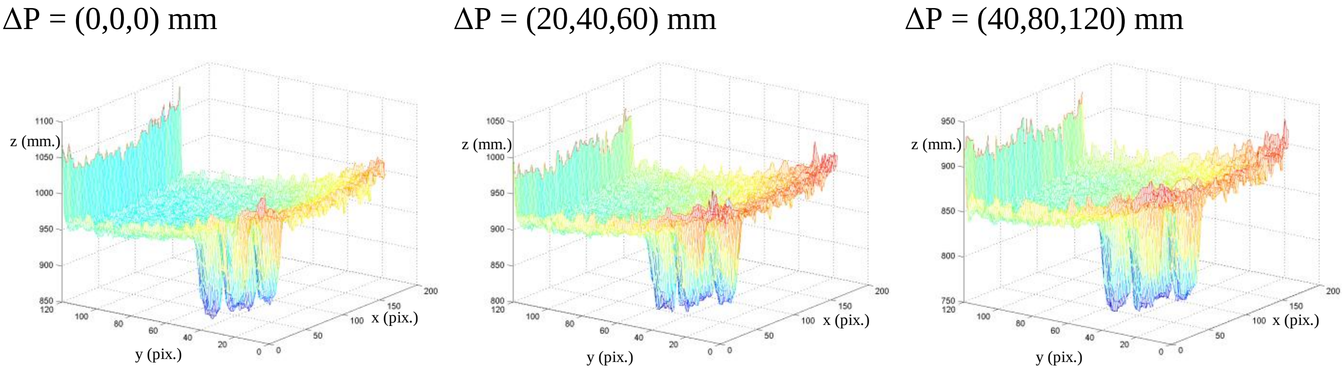


Figure 12. Range Image computed for the integration time updated at each iteration.


ΔP = (0,0,0) mm

ΔP = (20,40,60) mm

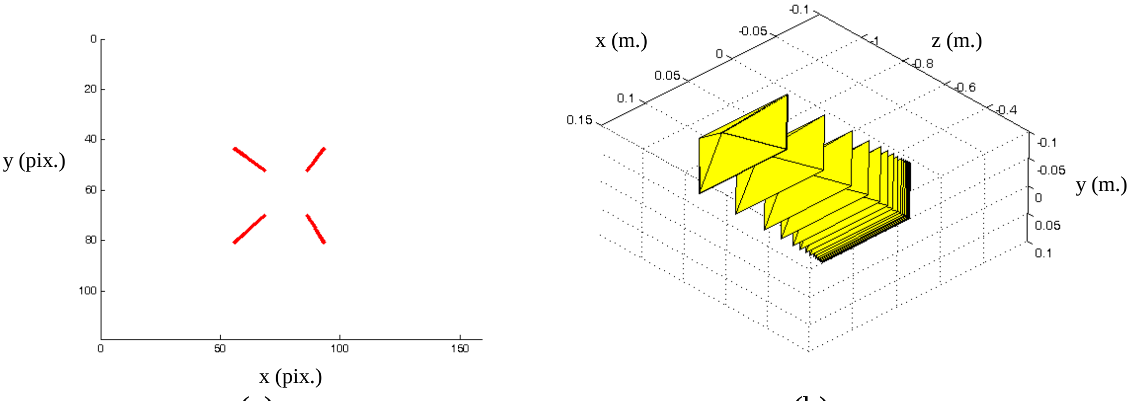
ΔP = (40,80,120) mm

The image ranges shown in Figure 11 are better than those in Figure 6 because the integration time has been updated during the visual servoing task. The distance between the camera and the target has changed as Figure 10 shows and the camera PMD19k has been self-configured with suitable integration time values. In this example, the integration times have been (53, 41 and 35 mseg).

6.2. Trajectory 2

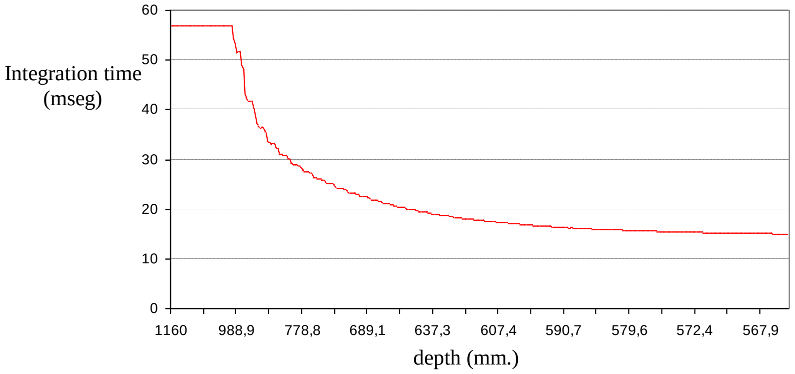
In this case, a trajectory with a displacement only in depth is described. The initial and final positions of the features in the image are (68,51)(86,51)(68,70)(86,70) and (56,43)(93,43)(56,80)(93,80), respectively. The initial distance between the eye-in-hand camera and the object is 1,160 mm and the final distance is 560 mm by using the proposed control law, the robot is able to perform precisely the displacement in depth as Figure 13 shows. In order to complete the task, the integration time has been updated using the algorithm described in Section 5 and thus the evolution represented in Figure 14 is obtained. As we have previously indicated [see Figure 2(b)], the minimum, $\tau_{\mathtt{min}}$, and maximum,$\tau_{\mathtt{max}}$, values of the integration time are 10 ms and 57.4 ms, respectively. Therefore, when the theoretical value for the integration time is greater than $\tau_{\mathtt{max}}$ this parameter is saturated to 57.4 ms (see Figure 14).

Figure 13. Trajectory during the visual servoing task. (a) Trajectory of the image features. (b) Trajectory of the eye-in-hand camera. Experiment 2.



(a) (b)

Figure 14. Integration time values at each iteration of the visual servoing task. Experiment 2.



6.3. Trajectory 3

As described in [1], in classical image-based visual servoing systems the depth of each image feature must be estimated at each iteration of the control scheme. In order to avoid the necessity of estimating these parameters, one popular approach is to choose $\hat{\mathbf{L}}_{\mathrm{s}}^{+} = \mathbf{L}_{\mathrm{s}^*}^{+}$, where $\mathbf{L}_{\mathrm{s}^*}$ is the value of $\mathbf{L}_{\mathrm{s}}$ for the desired position $\mathbf{s}^*$**.** In this case, $\mathbf{L}_{\mathrm{s}^*}^{+}$ is constant, an only the desired depth of each point has to be set, and thus, no varying 3D parameters have to be estimated during the visual servoing. In this section, a comparison between this last approach and the one proposed in this article is shown. To do so, a visual servo task is considered in which the initial position of the visual features in the image are (105,83)(119,73)(114,98)(130,89) and the desired position for the image features are (13,27)(44,20)(20,58)(51,51) [Figure 15(a)]. The initial and final positions of the eye-in-hand camera are represented in Figure 15(b).

Figure 15. (a) Initial position of the image features and the eye-in-hand camera. (b) Final position of the image features and the eye-in-hand camera. Experiment 3.

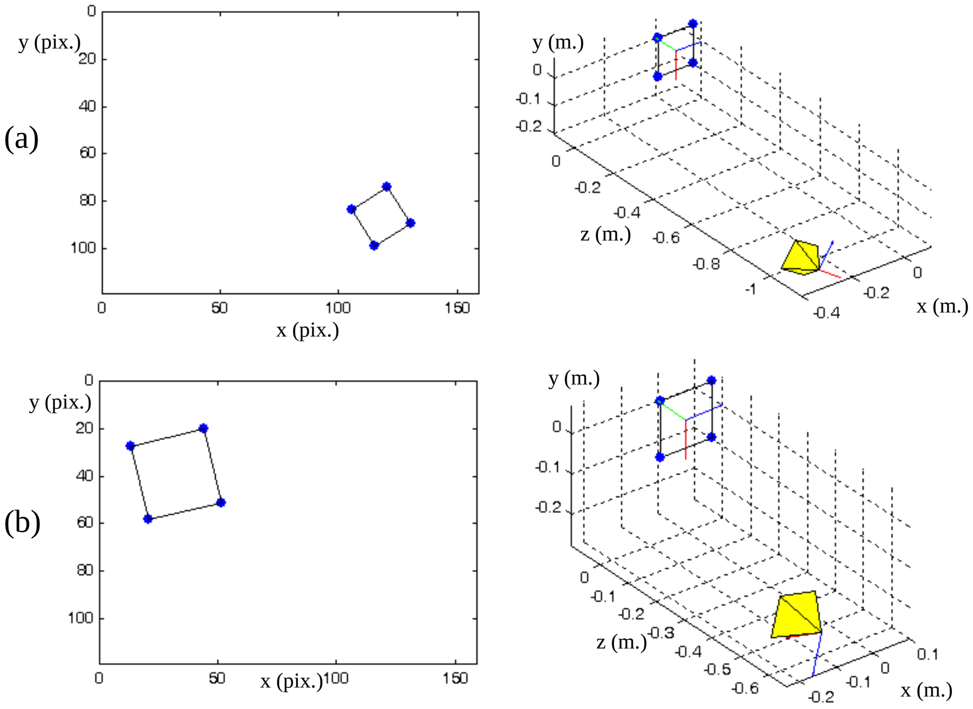


Figure 16 shows the evolution of the image features which are obtained when a classical image-based visual servoing system with $\hat{\mathbf{L}}_s^+ = \mathbf{L}_{s^*}^+$ is applied. In this case, the visual features are lost and the image features does not converge towards the desired ones. However, the use of the control law and the depth estimation proposed in Equations (9) and (10) generates the behaviour represented in Figure 17. In this last figure we can see that the visual servoing system is able to converge towards the desired location. This experiment shows the necessity of correctly estimating the depth parameters in order assure the correct convergence.

Figure 16. Image trajectory when $\hat{\mathbf{L}}_s^+ = \mathbf{L}_{s^*}^+$.

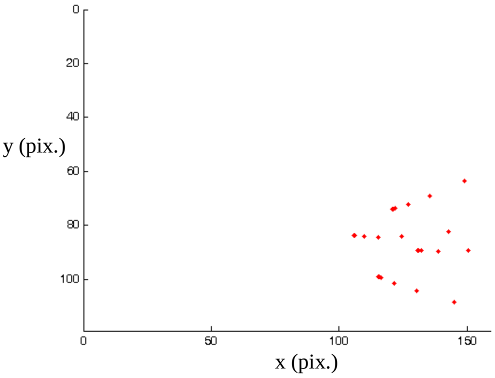
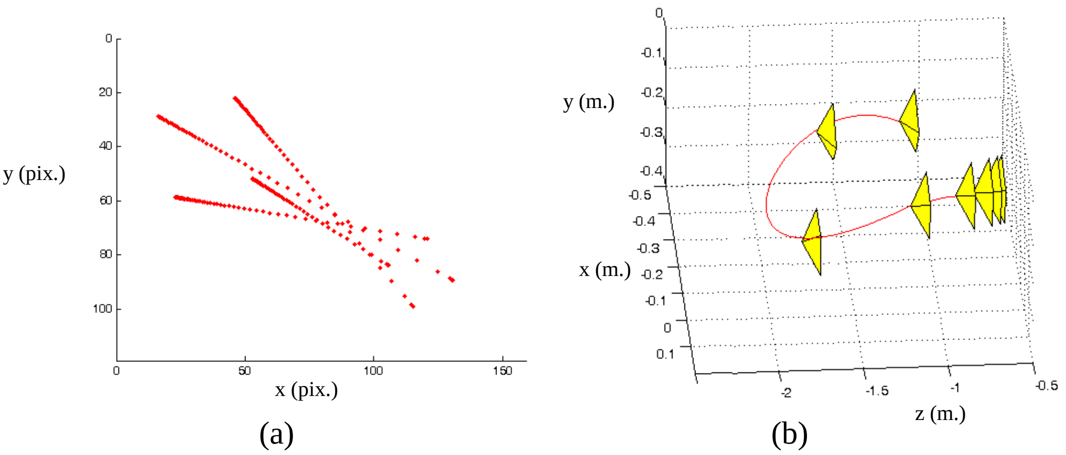


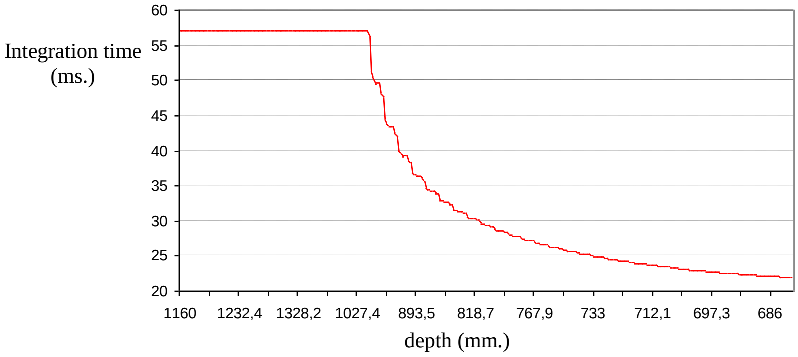
Figure 17. Trajectory during the visual servoing task. (a) Trajectory of the image features. (b) Trajectory of the eye-in-hand camera. Experiment 3.



(a) (b)

In this experiment there are important variations in the distance between the camera and the object from which the features are extracted. The initial and final depths are 1,160 mm. and 680 mm. respectively, and during the task the depth arrive until 1,760 mm. Thus, considering a fixed integration time, important errors appear and the task cannot be performed. Therefore, the integration time has to be updated with the approach described in this paper, and thus the evolution represented in Figure 18 is obtained. In this experiment the integration time is limited to values between the minimum, τ_{min}, and maximum,τ_{max}, in the same way that in the previous experiment according to Figure 2(b).

Figure 18. Integration time values at each iteration of the visual servoing task. Experiment 3.



7. Conclusions

This paper presents a new image-based visual servoing system which integrates range information in the interaction matrix. Another property of the proposed system is the possibility of performing the camera calibration during the task. To do this, the visual servoing system uses the range images not only to determine the depths of the object features but also to adjust the camera integration time during the task.

When a ToF camera is employed to guide a robot, the distance between the camera and the objects of the workspace change. Therefore, the camera integration time must be updated in order to correctly observe the objects of the workspace. As it is demonstrated in the experiments, the integration time must be updated depending on the distance between the camera and the objects. The use of the proposed approach guarantees that the information obtained from the ToF camera is accurate because an adequate integration time is employed at each moment. This last aspect permits obtaining a better estimation for the objects depth. Therefore, the behaviour of the visual servoing is enhanced with respect to previous approaches where this parameter is not accurately estimated. Currently, we are working in determining the accurate dynamic model of the robot to improve the visual servoing control law in order to assure the given specifications during the task.

Acknowledgements

The authors want to express their gratitude to the Spanish Ministry of Science and Innovation for their financial support through the project DPI2008-02647 and to the Research and Innovation Vicepresident Office of the University of Alicante for their financial support through the emergent projects.

References

1. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82-90.
2. de Jong, F.; Pieter P.J. Visual Servoing in PCB Manufacturing. In *Proceedings of the 6th Annual Conference of the Advanced School for Computing and Imaging* (*ASCI),* Lommel, Belgium, June 14-16, 2000; pp. 59-63.
3. Reiser, U.; Kubacki, J. Using a 3D Time-of-Flight Range Camera for Visual Tracking. In *Proceedings of 6th IFAC Symposium on Intelligent Autonomous Vehicles*. Toulouse, France, September 3–5, 2007.
4. Klank, U.; Pangercic, D.; Rusu, R.B.; Beetz, M. Real-time CAD Model Matching for Mobile Manipulation and Grasping. In *Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids),* Paris, France, December 7-10, 2009.
5. Fuchs, S.; Hirzinger, G. Extrinsic and Depth Calibration of ToF-cameras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008),* Anchorage, AK, USA, June 24–26, 2008; pp. 1-6.
6. Khongsab, P. *Signal Processing and Performance Evaluation of a PMD Camera for Space Docking*. Master Thesis. Luleå University of Technology, Luleå, Sweden, 2009.
7. Lindner, M.; Kolb, A. Calibration of the Intensity-Related Distance Error of the PMD TOF-Camera. In *Proceedings of SPIE XXV Conference on Intelligent Robots and Computer Vision,* Boston, MA, USA, September 15–17, 2007; pp. 6764-6771.
8. Wiedemann, M.; Sauer, M.; Driewer, F.; Schilling, K. Analysis and Characterization of the PMD Camera for Application in Mobile Robotics. In *Proceedings of the 17th IFAC World Congress*, Coex, Korea, July 6–11, 2008, pp. 13689-13694.
9. May, S.; Werner, B.; Surmann, H.; Pervölz, K. 3D Time-of-flight Cameras for Mobile Robotics. In *Proceedings of IEEE Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, October 9–15, 2006; pp. 790-795.

10. Mure Dubois, J.; Hugli, H. Fusion of Time of Flight Camera Point Clouds. In *Proceedings of Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, October 18, 2008.
11. Comport, A.I.; Marchand, E.; Pressigout, M.; Chaumette, F. Real-time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework. *IEEE Trans. Visual. Comput. Gr*. **2006**, *12*, 615-628.
12. Marchand, E.; Chaumette, F. Virtual Visual Servoing: A Framework for Real-time Augmented Reality. *Computer Graphics Forum*, **2002**, *21*, 289-298.
13. Benhimane, S.; Malis, E. Homography-based 2D Visual Tracking and Servoing. *Int. J. Robot. Res.* **2007**, *26*, 661-676.
14. Hadj-Abdelkader, H.; Mezouar, Y.; Martinet, P.; Chaumette, F. Catadioptric Visual Servoing from 3D Straight Lines. *IEEE Trans. Robot.* **2008**, *24*, 652-665.
15. Zhang, Z. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Corfu, Greece, September 20–27, 1999; Volume 1, pp. 666-673.
16. Rapp, H. *Experimental and Theorical Investigation of Correlating ToF Camera Systems*. Master Thesis. University of Heidelberg, Heidelberg, Germany, September 2007.
17. Gil, P; Pomares, J.; Torres, F. Analysis and Adaptation of Integration Time in PMD Camera for Visual Servoing. In *Proceedings of 20th International Conference on Pattern Recongnition (ICPR 2010)*, Istanbul, Turkey, August 2010.
18. Pomares, J.; Chaumette, F.; Torres, F. Adaptive visual servoing by simultaneous camera calibration. In *Proceedings of IEEE International Conference on Robotics and Automation*, Rome, Italy, April 10–14 2007; pp. 2811-2816.