European Journal of Control (2011)4:1–15

DOI:10.3166/EJC.17.1–15

European Journal of Control

Single Pan and Tilt Camera Indoor Positioning and Tracking System

Tiago Gaspar and Paulo Oliveira

IST/ISR, Lisboa, Portugal

An inexpensive single pan and tilt camera based indoor positioning and tracking system is proposed, supported on a functional architecture where three main modules can be identified: one related to the interface with the camera, tackled with parameter estimation techniques; other, responsible for isolating and identifying the target, based on advanced image processing techniques, and a third, that resorting to nonlinear dynamic system suboptimal state estimation techniques, performs the tracking of the target and estimates its position, and linear and angular velocities. The contributions of this work are fourfold: i) a new indoor positioning and tracking system architecture; ii) a new lens distortion calibration method, that preserves generic straight lines in images; iii) suboptimal nonlinear multiple-model adaptive estimation techniques, for the adopted target model, to tackle the positioning and tracking tasks, and iv) the implementation and validation in real time of a complex tracking system, based on a low cost single camera. To assess the performance of the proposed system, a series of indoor experimental tests for a range of operation of up to ten meter were carried out. A centimetric accuracy was obtained under realistic conditions.

Keywords: Indoor Positioning and Tracking, Nonlinear Filtering, Multiple-Model Adaptive Estimation, Single Camera Vision Systems.

1. Introduction

With the development and the widespread use of robotic systems, localization and tracking have become fundamental issues that must be addressed in order to provide autonomous capabilities to a robot. The availability of reliable estimates for a robot position is essential to its navigation and control systems, which justifies the significant effort that has been put into this domain, see [14], [2], and [4].

In outdoor applications, the NAVSTAR Global Positioning System (GPS) has been widely explored with satisfactory results for most of the actual needs. Indoor positioning systems based on this technology however face some undesirable effects, like multipath and strong attenuation of electromagnetic waves, precluding their use. Successfully exploited alternative techniques have been reported, such as infrared radiation, ultrasound, radio frequency, and vision [14]. A summarized description of these techniques is presented next.

Infrared (IR) based positioning systems use modulated IR light to transmit the identity of a mobile device to a fixed receiver in a known location. To estimate the position of this mobile platform, a set of IR receivers is distributed throughout the space in which the platform carries out its mission. However, this approach has low resolution, since it is impossible to determine the position of a target with higher resolution than the known locations of the base stations. Besides, it requires high installation costs and the existence of line of sight between the mobile device and the receiver. Moreover, the overall performance degrades under direct sun light or high ambient temperature. On the other hand, these systems are appropriate for spaces in which other technologies do not perform properly.

Ultrasound technology uses the time-of-flight of ultrasonic waves to compute the distance between a receiver,

Correspondence to: E-mail: pjcro@isr.ist.utl.pt

Received 12 September 2009; Accepted 22 September 2010
Recommended by G.R. Damm, E.F. Camacho

installed in a known position, and a target, to which the transmitter is attached. The precision of this technique is limited by the possibility of existence of multipaths between both devices, by the variation of ultrasonic waves velocity with humidity and environment temperature, and by the possible interference from other acoustic sources. Ultrasound based positioning systems are widely applied whenever the main purpose is to increase the accuracy of the system. However, this type of systems has high implementation costs, which make it inaccessible to most users.

There are several radio frequency (RF) based approaches that can be adopted. Systems supported in WLAN (Wireless Local Area Network), for instance, determine the distance between a sender and a receiver based on the strength of the received radio waves. This method reaches accuracies of approximately $1-3$ m, and is vulnerable to multipath, signal reflection, and diffraction. The multipath problem can be mitigated by using UWB (Ultra Wide Band) technology, which operates by emitting a series of extremely short pulses that improve the ability to correctly identify the original signal. This technique leads to accuracies of approximately $10-30$ cm, however implies an increase on the system cost and complexity.

The development of digital image processing techniques, associated with the widespread availability of embedded digital low power processors, has contributed to the enlargement of the domain of applicability of computer vision based positioning systems. Despite some undesirable visual effects that might difficult the identification of targets, the following properties contributed to the success of this kind of systems: the great diversity of information contained in each image, the fact that cameras are not vulnerable to interferences similar to the ones that affect the active sensors just described, and the possibility of pursuing several targets with just one camera. The attainable accuracies of vision based systems are very wide, and depend significantly on the resolution and cost of cameras used. The most expensive cameras, with high resolutions, achieve higher accuracies. However, the use of these cameras requires significant investments and the availability of high computacional power, to cope with the large amounts of collected data. The evolution of electronics and the presence of security cameras in almost every building have made vision based positioning systems one of the most successful solutions in this domain.

An alternative approach to the ones described before consists in combining different techniques, as occur in the successful *Active Bat* and *Cricket* systems [14], that use both ultrasound and RF measurements. The *Active Bat* system, for instance, can locate a target to within 9 cm of its true position for 95% of the measurements. However, these systems have a meaningful drawback, which is its high cost.

The indoor tracking system proposed in this work uses vision technology, since this technique has a growing domain of applicability and leads to interesting results with very low investments. This system uses a single camera and estimates in real time the position, velocity, and acceleration of a target that evolves along an unknown trajectory, in the 3D world, as well as its angular velocity. In order to accomplish this purpose, a new positioning and tracking system is detailed, based on suboptimal stochastic multiple-model adaptive estimation techniques. The complete process of synthesis, analysis, implementation, and validation in real time is presented next and builds on preliminary versions presented in [11] and [10]. A survey on 3D tracking systems can be found in [15], where a comprehensive study of the literature on this subject is also provided.

This document is organized as follows: in section 2, the architecture and the main methodologies and algorithms used in the positioning and tracking system proposed are described. In section 3, the camera and lens models are studied in detail. To isolate and identify the target, advanced image processing algorithms are discussed in section 4, and in section 5 a multiple-model nonlinear estimation technique is proposed. Section 6 analyzes the experimental results obtained, and in section 7 some concluding remarks are addressed.

2. System Architecture

In this work, a new indoor positioning system architecture is proposed, based on three main modules: one that addresses the interface with the camera, a second that implements image processing algorithms, and a third responsible for dynamic systems state estimation. The proposed architecture is presented in Fig. 1, where some quantities are introduced informally to augment the legibility of the document.

The extraction of physical information from an image acquired by a camera requires the knowledge of its intrinsic ($\mathbf{A}$) and extrinsic ($\mathbf{R}$ and $\mathbf{T}$) parameters, which are computed during the initial calibration process. In this work, calibration was preceded by an independent determination of a set of parameters ($\mathbf{K}$) responsible for compensating the distortion introduced by the lens of the camera. Since the low cost camera used has no orientation sensor, the knowledge of its position in each moment required the development of an external algorithm capable of estimating its instantaneous pan and tilt angles (α_r and θ_r, respectively).

The target identification is the main purpose of the image processing module. Active contours method, usually denominated snakes, was selected to track important features in each acquired image. The approach selected

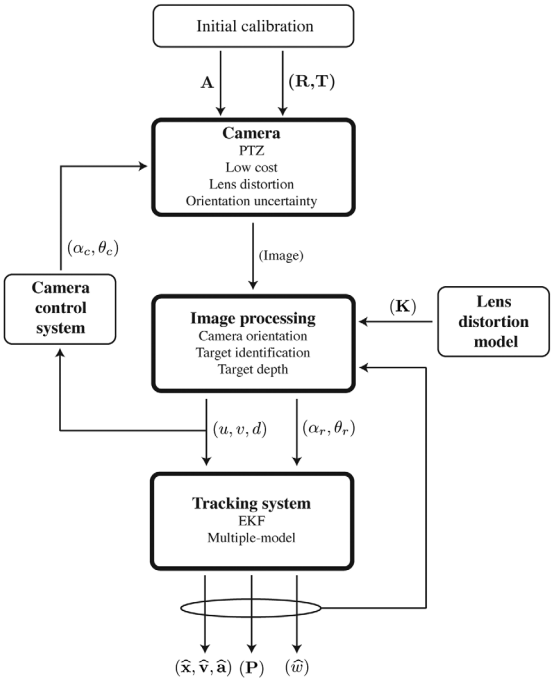


Fig. 1. Tracking system architecture.

consists in estimating the target contour, that provides the necessary information to compute the target center coordinates (u, v) and its distance (d) to the origin of the world reference frame. These quantities correspond to the measurements that are used to estimate the position $(\hat{\mathbf{x}})$, velocity $(\hat{\mathbf{v}})$, and acceleration $(\hat{\mathbf{a}})$ of the body to be tracked. Note that the computation of d requires the knowledge of the real dimensions of the target, since the proposed system uses one single camera instead of a stereo configuration.

To obtain estimates on the state and parameters of the underlying dynamic system, an estimation problem is formulated and solved. However, the dynamic model adopted and the sensor used have nonlinear characteristics. Extended Kalman filters included in a multiple-model adaptive estimation architecture were selected to provide estimates on the system state ($\hat{\mathbf{x}}$, $\hat{\mathbf{v}}$, and $\hat{\mathbf{a}}$), to identify the unknown target angular velocity $\hat{w}$, and to estimate the error covariance $\mathbf{P}$, as depicted in Fig. 1.

The command of the camera is the result of solving a decision problem with the purpose of maintaining the target close to the image center. Since the range of movements available is very restricted, the implemented decision system, which consists in computing the pan and tilt angles (α_c and θ_c, respectively) that should be sent to the camera at each moment, is very simple. Large distances between the target and the acquired images center are avoided. Thus, the capability of the overall system to follow targets is increased.

3. Sensor: PTZ Camera

In this section, camera and lens models are described. Moreover, the techniques selected to tackle the identification and calibration of the sensor are detailed.

3.1. Camera Model

Given the high complexity of the camera optical system, and the consequent high number of parameters required to model the whole image acquisition process, it is common to exploit a linear model to the camera. In this architecture the classical pinhole model was considered [7].

Let $\mathbf{M} = [x, y, z, t]^T$ be the homogeneous coordinates of a visible point in the world reference frame, and $\mathbf{m} = [u, v, s]^T$ the homogeneous coordinates of that point projected into the image frame, where t and s are the coordinates added when transforming the euclidean coordinates into homogeneous. According to this model, the relation between the coordinates expressed in the world and image frames is given by

$$\lambda \mathbf{m} = \mathbf{P}\mathbf{M}, \tag{1}$$

where λ is a multiplicative constant related with the distance from the point in space to the camera, and $\mathbf{P}$ is the projection matrix that relates 3D world coordinates and 2D image coordinates. The transformation given by this matrix can be decomposed into three others: one between world and camera coordinate frames, expressed by ${}^c\mathbf{g}_M$ in homogeneous coordinates; other responsible for projecting 3D points into the image plane, represented by π, and a third that changes the origin and units of the coordinate system used to identify each point in acquired images, denoted $\mathbf{A}$.

The transformation between world and camera coordinate frames can be obtained by a rigid body transformation

$$\mathbf{M}_c = {}^c\mathbf{g}_M \mathbf{M}_M, \qquad {}^c\mathbf{g}_M = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \in \Re^{4\times 4},$$

where $\mathbf{M}_M$ corresponds to the homogeneous coordinates of a point in the world reference frame, and $\mathbf{M}_c$ to its correspondent homogeneous coordinates in the camera reference frame. Matrix $\mathbf{R}$ is a rotation matrix belonging to SO(3), i.e. verifies $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ and $det(\mathbf{R}) = 1$; $\mathbf{T} \in \Re^3$ is a translation vector, and $\mathbf{0}$ is a null vector of dimension 1×3. The transformation parameters $\{\mathbf{R}, \mathbf{T}\}$ are the extrinsic parameters of the camera, since they only depend on its position and orientation with respect to the world reference frame.

The 3D to 2D transformation can be expressed in homogeneous coordinates by

$$z_c \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \pi \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \ \pi = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where (x_c, y_c, z_c) are the cartesian coordinates of a point in the camera coordinate frame, (x_p, y_p) are its correspondent coordinates in the image plane, and f is the focal length of the pinhole camera model, expressed in mm. Without loss of generality, f is assumed to be unitary in the world coordinate system ($f = 1$), leading to

$$\pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The transformation studied before considers a 2D coordinate system centered in the principal point (intersection of the optical axis with the image plane), whose coordinates (x_p, y_p) are measured in mm. However, in practical applications it is common to use a reference frame located on the image top left corner, with coordinates (u, v) measured in pixels. The relation between the two referred coordinate frames can be expressed in homogeneous coordinates by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix},$$

where u_0 and v_0 are the coordinates in pixels of the principal point, and α_u and α_v are conversion factors from mm to pixels. This transformation matrix is here denoted $\mathbf{A}$, and corresponds to the intrinsic parameters matrix, since its elements depend on the internal properties of the camera (such as its zoom level, for instance), but not on its orientation and/or position in the world coordinate system.

The product of the three previous transformations leads to the $\mathbf{P}$ overall expression, $\mathbf{P} = \mathbf{A}.\pi.{}^{c}\mathbf{g}_{M}$, that establishes the relation between a point in the 3D world and its projection into acquired images. The use of this model requires the determination of the camera intrinsic and extrinsic parameters. In this work, the classical approach proposed by Faugeras [7] was selected and implemented, with the major advantages that only one image is required and reliable results can be obtained. An independent algorithm that compensates for lens distortion was implemented, as described in section 3.3.

3.2. PTZ Camera Internal Geometry

The camera used in this work has the ability to describe pan and tilt movements, which leads to the variation of its extrinsic parameters over time. Therefore, the rigorous definition of the rigid body transformation between camera and world reference frames requires the adoption of a model for the camera internal geometry and the study of its direct kinematics. The *Creative WebCam Live! Motion* camera used has closed architecture, thus its internal geometry model was based on the analysis of its external structure, and the parameters of this model were computed from a small number of experiments.

The proposed model includes five transformations: one between the world reference frame and frame 0, whose origin coincides with the rotation center of the camera; three related to pan, tilt, and roll rotation movements, that take place according to this order, and that give the transformation between frames 0 and 3, and a fifth one between the resulting frame of the previous transformations and the camera reference frame (whose origin coincides with its optical center). Given its inability to realize roll movements, the camera can be placed in any position and with any orientation in relation to the world coordinate system, since the roll degree of freedom was included in the model. This model considers that the camera optical and rotation centers are aligned with exception of an offset in the optical axis direction, which is plausible given its external geometry.

Previous transformations can be expressed as

$${}^{M}\mathbf{g}_0 = \begin{bmatrix} 1 & 0 & 0 & {}^{M}Px_c - \delta \\ 0 & 1 & 0 & {}^{M}Py_c \\ 0 & 0 & 1 & {}^{M}Pz_c \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^{0}\mathbf{g}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) & 0 \\ 0 & \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^{1}\mathbf{g}_2 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^{2}\mathbf{g}_3 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\theta) & -\cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and}$$

$${}^{3}\mathbf{g}_c = \begin{bmatrix} 0 & 0 & 1 & \delta \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where α, θ, and ψ correspond to the pan, tilt, and roll angles, respectively, δ is the offset between

the camera optical and rotation centers, and ${}^{M}\mathbf{P}_c = \left[{}^{M}Px_c, {}^{M}Py_c, {}^{M}Pz_c\right]^T$ represents the optical center coordinates in the world reference frame, when $\alpha = \theta = \psi = 0^o$.

The composition of these transformations leads to the global transformation between world and camera reference frames:

$$ {}^{c}\mathbf{g}_M = {}^{M}\mathbf{g}_c^{-1}, \qquad {}^{M}\mathbf{g}_c = {}^{M}\mathbf{g}_0{}^{0}\mathbf{g}_1{}^{1}\mathbf{g}_2{}^{2}\mathbf{g}_3{}^{3}\mathbf{g}_c, $$

that is fundamental to determine the camera projection matrix over time.

Expressions introduced require the knowledge of five parameters: pan, tilt, and roll angles, the position of the camera optical center in the world coordinate frame, when these angles are zero, and the offset between this point and the camera rotation center. Given the high uncertainty associated with the pan and tilt movements of the camera used, there is no way of accurately setting its orientation to $\alpha = \theta = \psi = 0^o$. Therefore, the determination of ${}^{M}\mathbf{P}_c$ was based upon interpolation methods. The approach adopted consisted in calibrating the camera for different orientations in relation to world reference frame, and, from the obtained angles and optical center coordinates, estimate the coordinates of the optical center when $\alpha = \theta = \psi = 0^o$, by means of a linear interpolation.

The determination of the offset between the camera optical and rotation centers (δ) was performed resorting to four points with known coordinates in the world. Calibrating the camera in different orientations, and writing the pair of equations

$$ u\mathbf{P}_3\mathbf{M} - \mathbf{P}_1\mathbf{M} = 0 $$
$$ v\mathbf{P}_3\mathbf{M} - \mathbf{P}_2\mathbf{M} = 0, $$

as a function of δ, for each one of the referred points, where $\mathbf{P}_i$ corresponds to the i-th line of each obtained projection matrix, yields $\mathbf{A}\delta = \mathbf{b}$, where matrices $\mathbf{A}$ and $\mathbf{b}$ result from the collected data. The value of the offset between the camera optical and rotation centers can then be computed resorting to the least mean squares method.

Camera orientation is determined in real time using reference points in the 3D world, whose image coordinates should be set on an initial stage. These points must be identified in each acquired image. The orientation of the camera can be obtained comparing the relative orientation of both i) the axis that connects the camera optical center to those points in the image, and ii) the camera optical axis, obtained as an output of the calibration process. Note that this procedure applies only to the camera pan and tilt angles, since these are the two unique degrees of freedom (ψ sticks to the value estimated in the calibration process).

In the approach adopted, the camera optical and rotation centers were considered coincident, what leads to a computationally more efficient algorithm to estimate the camera orientation, and, as a consequence, more suitable to real time applications. Besides, the influence of this approximation can be minimized by using reference points placed at large distances from the camera, when compared with the few millimeters that separate both centers.

3.3. Lens Distortion

The mapping function of the pinhole camera between the 3D world and the 2D camera image is linear, when expressed in homogeneous coordinates. However, if a low-cost or wide-angle lens system is used, the linear pinhole camera model fails. In those situations, and for the camera used in this work, the radial lens distortion is the main source of errors and no vestige of tangential distortion was identified. Therefore, it is necessary to compensate for this distortion using a nonlinear inverse radial distortion function, which corrects measurements in the 2D camera image to those that would have been obtained with an ideal linear pinhole camera model.

The inverse radial distortion function is a mapping that recovers the coordinates (x, y) of undistorted points from the coordinates (x_d, y_d) of the correspondent distorted points, where both coordinates are related to a reference frame with origin in the image distortion center (x_0, y_0). Since radial deformation increases with the distance to the distortion center, the inverse radial distortion function $f(r_d)$ can be approximated and parameterized by the following Taylor expansion:

$$ r = f(r_d) = r_d + r_d \sum_{i=0}^{\infty} k_i r_d^{i-1}, $$

with $r = \sqrt{x^2 + y^2}$ and $r_d = \sqrt{x_d^2 + y_d^2}$, that results in

$$ x = x_d + x_d \sum_{i=0}^{\infty} k_i r_d^{i-1} \quad \text{and} \quad y = y_d + y_d \sum_{i=0}^{\infty} k_i r_d^{i-1}. $$

According to practical tests, only the parameters k_3 and k_5 are required to obtain good results, similarly with the conclusions in [22]. Using more parameters brings no significative improvement to the approximation of $f(r_d)$, for images in video resolution, and an estimation of less parameters is more robust.

There are two main approaches that are commonly used to estimate the camera radial distortion function: one that includes this procedure in the calibration process responsible for determining the pinhole model parameters, and other that addresses these two problems independently. The lens distortion compensation method that we propose in the remainder of this section explores the later idea, which allows to model any camera with the pinhole

model after the application of the inverse of the distortion function to image features.

The method proposed is based on the *rationale* that straight lines in the 3D space must remain straight lines in 2D images. This idea was already explored in the past, e.g. [22] and [6], however, what is proposed here is a simpler method that consists of two main steps: identification of image lines that correspond to straight lines in the scene, and estimation of the parameters of the inverse radial distortion function.

In contrast to other line-based approaches, that put a great effort into identifying straight lines in real world scenarios, our method assumes that segmentation of straight lines is performed accurately, what is reasonable since a scene where the identification of straight lines is straightforward can be easily created. This approach has no major impact on the applicability of the method, since this calibration is performed just once for each camera, and limits the influence of outliers in the overall algorithm performance, which otherwise could be tackled resorting to RANSAC method [9]. Moreover, it allows to distribute straight lines in such a way that some of them project into regions of the image that best capture the lens distortion characteristics, i.e. regions where the deformation is greater.

Ideally, if acquired images were not affected by distortion, 3D world straight lines would be preserved in 2D images. Therefore, the estimation of the inverse radial distortion model parameters was based on the resolution of the following set of equations

$$\begin{cases} f_{i1} = (y_{i1} - \widehat{y}_{i1}(m_i, b_i, x_{i1}))^2 = 0 \\ \qquad\qquad \vdots \\ f_{iN_p} = (y_{iN_p} - \widehat{y}_{iN_p}(m_i, b_i, x_{iN_p}))^2 = 0, \end{cases}$$
$$i = 1, \ldots, N_r,$$

with $\widehat{y}_{ij}(m_i, b_i, x_{ij}) = m_i x_{ij} + b_i$, where N_r and N_p are the number of straight lines and points per straight line acquired from the distorted image, respectively; (x_{ij}, y_{ij}) correspond to the j-th point of the i-th straight line estimated coordinates, given by the considered inverse distortion model, and $\widehat{y}_{ij}(m_i, b_i, x_{ij})$ corresponds to the y coordinate of the i-th straight line j-th point, given by the estimated slope-intercept equation of this straight line. A set of $N_r N_p$ nonlinear equations results, with a solution that can be found resorting to Newton's method; thus, estimates for parameters k_3, k_5, x_0, y_0, m_i, b_i, $i = 1, \ldots, N_r$ are obtained. Note that since intensity of radial distortion depends upon the distance to image distortion center, straight lines provided to the algorithm must be selected from different areas of the image, in order to avoid the polarization of the estimated parameters by image local properties.

Fig. 2. Inverse distortion model performance. Left: acquired (distorted) image; right: undistorted image.

The results obtained with the proposed distortion compensation method applied to a real image are presented in Fig. 2. Some straight lines in the 3D world turned into curves in the acquired image, however its original shape was retrieved by the method implemented.

4. Image Processing

In this section, image processing algorithms to implement the target isolation and identification are described, leading to the measurements to be provided to the estimation system.

4.1. Target Isolation and Identification

Every tracking system requires a mechanism to detect the target in acquired images. There are several strategies that can be adopted, being the most common based on point detectors, background subtraction, supervised learning, and segmentation algorithms such as mean-shift clustering and active contours, see [23] for details about these methodologies.

In this work, the isolation and identification of the target are tackled resorting to active contours. Active contours [13], or snakes, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. Snakes are widely used in several computer vision domains, such as edge detection [13], image segmentation [16], shape modeling [21], [19], or motion tracking [16], as happens in this application.

There are two main types of active contours models: *parametric active contours* [13] and *geometric active contours* [5]. In this work, the first type is used, in which a parameterized curve evolves over time towards the desired image features, usually edges, attracted by external forces given by the negative gradient of a potential function. These forces interact with internal ones responsible for

holding the curve together and keep it from bending too much.

Snakes are curves $\mathbf{x}(s) = [x(s), y(s)]$, $s \in [0, 1]$, that evolve through the spacial domain of an image seeking to minimize its energy

$$E_{sk} = E_{int} + E_{ext},$$

that includes a term related to its internal energy E_{int}, which has to do with its smoothness, and a term of external energy E_{ext}, based on forces extracted from the image. Traditionally, this energy can be expressed in the form

$$E_{sk} = \int_0^1 \frac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] + E_{ext}(\mathbf{x}(s))ds, \quad (2)$$

where the parameters α and β control the snake tension and rigidity, respectively, and $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ denote the first and second derivatives of $\mathbf{x}(s)$ with respect to s.

There are several typical methods of designing the external energy of an image that would lead a snake towards the desired features [13]. In this work, the external energy expression

$$E_{ext}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2$$

was considered, where $I(x, y)$ represents image intensity at coordinates (x, y); $G_\sigma(x, y)$ represents a 2D Gaussian function with standard deviation σ, and ∇ is the gradient operator. Although large values of σ blur image boundaries, such values increase the capture range of the active contour.

A curve $\mathbf{x}(s)$ that minimizes (2) must satisfy the Euler equation

$$\alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s) - \nabla E_{ext} = 0, \quad (3)$$

which is verified when the internal and external forces, $\mathbf{F}_{int} = \alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s)$ and $\mathbf{F}_{ext}^{(p)} = -\nabla E_{ext}$, respectively, reach an equilibrium.

In order to solve (3), consider the snake $\mathbf{x}$ also as a function of time t, and let its evolution be governed by its partial derivative with respect to t

$$\mathbf{x}_t(s, t) = \alpha\mathbf{x}''(s, t) - \beta\mathbf{x}''''(s, t) - \nabla E_{ext}, \quad (4)$$

that corresponds to the first member of (3). A solution of (3) is achieved when $\mathbf{x}_t(s, t)$ vanishes and, as a consequence, $\mathbf{x}(s, t)$ stabilizes. Note that the energy of the snake may not be a convex function, thus being possible the existence of local minima that lead the snake towards boundaries other than the desired ones. However, the impact of this limitation can be significantly minimized by initializing the snake in the neighborhood of the features of interest.

Approximating the derivatives in (4) by the spacial finite differences method, with step h, yields

$$(\mathbf{x}_t)_i = \frac{\alpha}{h^2}(\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}) - \frac{\beta}{h^4}(\mathbf{x}_{i+2} - 4\mathbf{x}_{i+1} + 6\mathbf{x}_i - 4\mathbf{x}_{i-1} + \mathbf{x}_{i-2}) + \mathbf{F}_{ext}^{(p)}(\mathbf{x}_i),$$

where $\mathbf{x}_i = \mathbf{x}(ih, t)$, and $\mathbf{F}_{ext}^{(p)}(\mathbf{x}_i)$ represents the image influence at the point $\mathbf{x}_i$.

The temporal evolution of the active contour in the image domain occurs according to $\mathbf{x}^{n+1} = \mathbf{x}^n + \tau\mathbf{x}_t^n$, where τ is the time step. The iterative process ends when the coordinates of each point of the snake remain approximately constant over time.

4.2. Sensor Measurements

Once defined the target contour identification procedure, it is important to make a brief overview on the way this information is used. The measurements that will be provided to the estimation process are the target center coordinates (u, v) and its distance (d) to the origin of world reference frame.

Target center coordinates in each acquired image are computed easily from its estimated contour, as being the mean of the coordinates of the points that belong to this contour. Target distance to the origin of world reference frame is computed from its estimated boundary and its real dimensions in the 3D world. These dimensions and the knowledge of the camera intrinsic and extrinsic parameters allow to establish metric relations between image and world quantities. Estimates on the depth of the target can then be obtained.

The use of triangulation methods for at least two cameras would not require further knowledge on the target to compute its depth. However, the present tracking system uses a single camera. Thus, additional information must be available. In this work, it is assumed that target dimensions are known.

5. Tracking System

In this section, the implementation of nonlinear estimation methods is described. Estimates on the target position, velocity, and acceleration in the 3D world are provided, and the target angular velocity is identified. This estimator is based on measurements from the previously computed target center coordinates and distance to the origin of world reference frame.

5.1. Extended Kalman Filter

Kalman filter (KF) [12] provides an optimal solution to the problem of estimating the state of a discrete time process that is described by a linear stochastic difference equation. However, this approach is not valid when the process and/or the measurements are nonlinear. One of the most successful approaches, in these situations, consists in applying a linear time-varying Kalman filter to a system that results from the linearization of the original nonlinear one, along the estimates. This kind of filters are usually referred to as Extended Kalman filters (EKF) [12], and have the advantage of being computationally efficient, which is essential in real time applications.

Consider a nonlinear system with state $\mathbf{x} \in \Re^n$ expressed by the nonlinear stochastic difference equation

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}),$$

and with measurements available $\mathbf{z} \in \Re^m$ given by

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \tag{5}$$

where index k represents time, $\mathbf{u}_k$ the control input, and $\mathbf{w}_k \in \Re^n$ and $\mathbf{v}_k \in \Re^m$ are random variables that correspond to the process and measurement noise, respectively. These variables are assumed to be zero mean Gaussian probability density functions with covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. Furthermore, they are assumed to be independent, i.e. $E[\mathbf{w}_k {\mathbf{v}_k}^T] = 0$.

Adopting a standard notation, where $\widehat{\mathbf{x}}_k^-$ is the *a priori* state estimate at step k given the estimated state at step $k-1$; $\widehat{\mathbf{x}}_k$ is the *a posteriori* state estimate at step k (given measurement $\mathbf{z}_k$); ${\mathbf{P}_k}^-$ is the *a priori* estimate error covariance; and $\mathbf{P}_k$ is the *a posteriori* estimate error covariance, and starting from initial estimates for $\widehat{\mathbf{x}}_{k-1}$ and $\mathbf{P}_{k-1}$, the state and error covariance estimates evolution over time is given by the following equations, see [12] and references therein for details:

Predict step

$$\widehat{\mathbf{x}}_k^- = f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0)$$
$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T$$

Update step

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1}$$
$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\widehat{\mathbf{x}}_k^-, 0))$$
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-,$$

where $\mathbf{K}_k$ is the Kalman gain; $\mathbf{A}_k$ is the Jacobian matrix of partial derivatives of f with respect to $\mathbf{x}$; $\mathbf{W}_k$ is the Jacobian matrix of partial derivatives of f with respect to $\mathbf{w}$; $\mathbf{H}_k$ is the Jacobian matrix of partial derivatives of h with respect to $\mathbf{x}$; and $\mathbf{V}_k$ is the Jacobian matrix of partial derivatives of h with respect to $\mathbf{v}$.

In the case of linear dynamic systems, the estimates provided by the Kalman filter are optimal, in the sense that the mean square estimation error is minimized. Estimates computed by EKF are suboptimal. It is even possible that it does not converge to the system state in some situations. However, the good performance observed in many practical applications made this strategy the most successful and popular in nonlinear estimation.

The implementation of an EKF requires a mathematical model to the target and sensors used. The choice of appropriate models is extremely important since it improves significantly the target tracking system performance, reducing the effects of the limited observation data available in this kind of applications. Given the movements expected for the targets to be tracked, the 3D *Planar Constant-Turn Model*, as presented in [17], was selected. This model considers the vector $\mathbf{x} = [\mathsf{x}, \dot{\mathsf{x}}, \ddot{\mathsf{x}}, \mathsf{y}, \dot{\mathsf{y}}, \ddot{\mathsf{y}}, \mathsf{z}, \dot{\mathsf{z}}, \ddot{\mathsf{z}}]^T$ as the state of the target, where $[\mathsf{x}, \mathsf{y}, \mathsf{z}]$, $[\dot{\mathsf{x}}, \dot{\mathsf{y}}, \dot{\mathsf{z}}]$, and $[\ddot{\mathsf{x}}, \ddot{\mathsf{y}}, \ddot{\mathsf{z}}]$ are the target position, linear velocity, and linear acceleration in the 3D world, respectively.

According to this model, the target state dynamics is given by

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &= diag[\mathbf{F}(w), \mathbf{F}(w), \mathbf{F}(w)]\mathbf{x}_{k-1} + \mathbf{w}_{k-1}\end{aligned}$$
$$\mathbf{F}(w) = \begin{bmatrix} 1 & \frac{sin(wT)}{w} & \frac{1-cos(wT)}{w^2} \\ 0 & cos(wT) & \frac{sin(wT)}{w} \\ 0 & -wsin(wT) & cos(wT) \end{bmatrix}, \tag{6}$$

where T is the sampling interval and w the target unknown angular velocity.

The absence of the control input $\mathbf{u}$ from the previous expression is due to its unknown nature. In fact, this is an important source of uncertainty in target tracking, and constitutes one of the major challenges in the positioning systems domain.

According to the same model, the process noise covariance matrix is given by $\mathbf{Q}(w) = diag[S_x\mathbf{U}(w), S_y\mathbf{U}(w), S_z\mathbf{U}(w)]$, where $diag[S_x, S_y, S_z]$ corresponds to its power spectral density matrix and

$$\mathbf{U}(w) = \begin{bmatrix} \frac{6wT - 8\sin(wT) + \sin(2wT)}{4w^5} & \frac{2\sin^4(wT/2)}{w^4} & \frac{-2wT + 4\sin(wT) - \sin(2wT)}{4w^3} \\ \frac{2\sin^4(wT/2)}{w^4} & \frac{2wT - sin(2wT)}{4w^3} & \frac{sin^2(wT)}{2w^2} \\ \frac{-2wT + 4\sin(wT) - \sin(2wT)}{4w^3} & \frac{sin^2(wT)}{2w^2} & \frac{2wT + \sin(2wT)}{4w} \end{bmatrix}.$$

The characterization of the whole system state dynamics requires yet the definition of matrix $\mathbf{W}$, that corresponds to the identity matrix of dimensions 9×9, as follows directly

form its definition and from the stochastic difference equation presented in (6).

Note that the presented dynamics and the process noise covariance matrix depend explicitly upon the target angular velocity. Since this information is not available, its identification will be tackled resorting to a multiple-model adaptive estimation approach (MMAE), as detailed next. An option would be to add the angular velocity (w) to the state vector, however this would increase its dimension and would make the model, that is linear in the state, highly nonlinear.

Sensor measurements available in each time instant, that define function $h(\mathbf{x}_k, \mathbf{v}_k)$, correspond to the target center coordinates (u, v) and target distance (d) to the origin of world reference frame, and are given by

$$\begin{aligned} u &= \frac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} + v_u \\ v &= \frac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} + v_v \\ d &= \sqrt{x^2 + y^2 + z^2} + v_d, \end{aligned} \tag{7}$$

where p_{ij} is the projection matrix element in line i and column j, and $\mathbf{v} = [v_u, v_v, v_d]^T$ is the measurement noise (the time step subscript k was omitted for simplicity of notation). The measurement vector is given by $\mathbf{z} = [u, v, d]^T$.

As can be inferred from (7), the measurement equation in (5) is nonlinear, which justifies the implementation of the EKF. The linearization of this function leads to matrix $\mathbf{H}$, with dimensions 3×9, and to matrix $\mathbf{V}$, that corresponds to the identity matrix of dimensions 3×3. Both matrices follow directly form its definition and the sensor measurements expression (7).

The complete measurement process characterization requires also the definition of the measurement noise covariance matrix $\mathbf{R}$. This matrix can be obtained from an accurate study of the available sensors, which consisted in executing a set of experiments aiming to compute the standard deviation of the estimation error in the image coordinates of a 3D world point, and the standard deviation of the error in the target depth estimation.

5.2. Multiple Models

The model considered for the target requires the knowledge of its angular velocity. However, this value is not known in real applications, which led us to the adoption of a multiple-model based approach that simultaneously identifies some parameters of the system and estimates its state. There are several architectures that resort to multiple models, either from a control perspective or from an estimation point of view.

In terms of control, most multiple-model approaches identify the most likely model by means of a "supervisor" that uses mainly deterministic concepts to decide which controller, from a bank of alternative candidates, must be switched into the feedback loop. These techniques are known as supervisory control techniques, e.g. [20]. An alternative control strategy consists in using stochastic methods to compute online *a posteriori* hypothesis probabilities reflecting which model is closest to the reality, see [8] for details.

In this work, a dual approach based upon an estimation perspective was adopted. The method implemented, known as *Multiple-Model Adaptive Estimator (MMAE)* [18], considers several models for a system that differ in a set of parameters (in this case the target angular velocity). Each one of these models includes an Extended Kalman filter, whose state estimates are mixed properly. The individual estimates are combined using a weighted sum with the *a posteriori* hypothesis probabilities of each model as weighting factors.

The *a posteriori* hypothesis probability of model $i = 1, \ldots, N$, where N corresponds to the number of considered models, evolves over time, from an initial estimate p_0^i, according to the following expression (see [18] for details):

$$p_k^i = \frac{\beta_k^i e^{-\frac{1}{2}\omega_k^i}}{\sum_{j=1}^{N} \beta_k^j e^{-\frac{1}{2}\omega_k^j} p_{k-1}^j} p_{k-1}^i,$$

with

$$\begin{aligned} \beta_k^i &= \frac{1}{(2\pi)^{m/2}\sqrt{det(\mathbf{S}_k^i)}} \\ \omega_k^i &= (\mathbf{r}_k^T)^i (\mathbf{S}_k^{-1})^i \mathbf{r}_k^i \end{aligned}$$

and

$$\begin{aligned} \mathbf{r}_k^i &= \mathbf{z}_k - \widetilde{\mathbf{z}}_k^i \\ \mathbf{S}_k^i &= \mathbf{H}_k^i (\mathbf{P}_k^-)^i (\mathbf{H}_k^T)^i + \mathbf{R}_k, \end{aligned}$$

where $\mathbf{r}^i$ is the residual vector of the i-th Kalman filter (difference between the sensor measurements, $\mathbf{z}_k$, and the ones predicted by model i, $\widetilde{\mathbf{z}}_k^i$), $\mathbf{S}^i$ is the residual covariance matrix associated with the i-th Kalman filter, m is the number of measurements (number of elements of $\mathbf{z}_k$), and k represents the time instant.

From the individual state estimates of each model, its error covariance matrices, and the *a posteriori* probability of each hypothesis, it is possible to compute the weighted

state estimate

$$\widehat{\mathbf{x}}_k = \sum_{j=1}^{N} p_k^j \widehat{\mathbf{x}}_k^j,$$

and the global covariance matrix

$$\mathbf{P}_k = \sum_{j=1}^{N} p_k^j [\mathbf{P}_k^j + (\widehat{\mathbf{x}}_k^j - \widehat{\mathbf{x}}_k)(\widehat{\mathbf{x}}_k^j - \widehat{\mathbf{x}}_k)^T].$$

5.3. Designing the Bank of EKFs

In this section, an insight into how to design the bank of Extended Kalman filters used in the MMAE architecture is given.

The use of multiple model approaches requires the definition of a criterion to divide the parameter set into smaller parameter subsets. In [8], for instance, this problem was addressed from the control point of view, and this division and the specification of the number of models used was based upon the definition of performance requirements for the resultant controller. Once determined the number of models to use, the subsets associated with each model must be computed, as well as the nominal angular velocity values of each Extended Kalman filter. With this purpose, the Baram Proximity Measure (BPM) is usually adopted, but techniques based upon the Kullback information metric can also be found in literature, see [3] and [1], respectively, for details.

Consider that w^* denotes the real value of the unknown parameter and that w^i denotes the nominal value used to implement the i-th Kalman Filter. If $w^* = w^i$, the steady-state KF residual $\mathbf{r}^*$ would be stationary white-noise with covariance matrix $\mathbf{S}^*$. On the other hand, if $w^* \neq w^i$, the residual $\mathbf{r}^i$ would not be white. The BPM is a function that measures the "stochastic distance" between residuals $\mathbf{r}^*$ and $\mathbf{r}^i$, and can be computed using the expression

$$L_*^i \equiv log|\mathbf{S}^i| + tr\{(\mathbf{S}^i)^{-1}\Gamma_*^i\}, \tag{8}$$

where L_*^i denotes the BPM between the i-th filter and the filter based upon the true model; $|\mathbf{S}^i|$ denotes the determinant of the steady-state prediction error covariance of the residual of the i-th filter; and Γ_*^i denotes the actual steady-state prediction error covariance of the residual of the i-th filter computed using information about the true model. A detailed deduction of (8) and description of the use of the BPM in multiple model architectures can be found in [3] and [8].

In order to find the number of models to use and the corresponding nominal parameter values, let the null angular velocity, that corresponds to a straight trajectory, be the nominal parameter of one of the models. Then search the remaining parameter set for the angular velocity nominal values that lead to a situation in which there is always a filter whose BPM, in relation to the filter based upon the true model, does no exceed a certain value. The boundaries of each subset are defined by the points of intersection of the BPM curves. For the system proposed, a total of $N = 4$ models results, with the nominal angular velocity values presented in the sequel.

According to the *fundamental convergence result*, proved in [3] for an arbitrary number of stable KFs, if the BPM from the true model to one of the nominal models is smaller than its BPM to any other model, then under some additional stationarity conditions and ergodicity assumptions the *a posteriori* probabilities will converge *almost surely* to the correct model (see [3], [8], and references therein for formal proofs and precise definitions of these concepts).

The system presented in this work is nonlinear, which invalidates the direct use of the BPM approach to design the bank of nonlinear filters proposed in previous sections. However, for performance study purposes, this technique can be used in a linearized version of the system under consideration. Therefore, by linearizing the sensor measurements expression presented in (7) about several different positions of the target in the world, and by using the BPM approach to design the bank of linear filters that result in each case, it is possible to gain some insight into how to choose the angular velocity nominal values that will be used in each EKF. However, it is important to stress that: i) as expected, this approach does not provide theoretical guarantees of convergence for the correct model, and ii) the linearizations mentioned above were only used in this process to help finding the nominal angular velocity values associated with each EKF.

Fig. 3 depicts the BPM for each one of the 4 models used. A linearized version of (7) for five different target depth nominal values was considered. The subset of

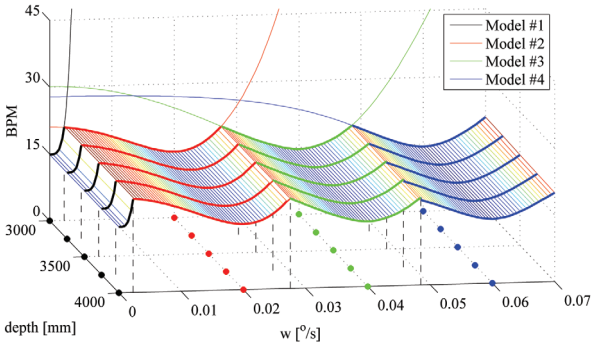


Fig. 3. BPM for the four models linearized about five different target positions, each one associated with a different target depth (noise conditions equal to the ones found in practice). Dots in different colours correspond to the angular velocity values that minimize the BPM in each subset.

validity of the first model is smaller than the remaining, which is intuitive since it corresponds to the identification of the particular situation in which the target is moving along a straight line. Another interesting result is that targets with angular velocities on the order of $2\pi 0.0075$ rads/s, for instance, would be identified by the second model and not by the first, which has the closest angular velocity value in terms of Euclidean distance. According to Fig. 3, models based upon the linearizations suggest that the BPM is insensitive to the position of the target, and that equally spaced angular velocity values are appropriate for the nominal values of the 4 models. The four angular velocity nominal values used in each EKF are the ones that minimize the BPMs presented in Fig.3, $2\pi\frac{1}{50}[0,1,2,3]$ rad/s, which lead to the division of the original set ($\Omega = [0, 2\pi 0.070]$ rad/s) into the following subsets:

$$\Omega_1 = 2\pi[0, 0.0023]\,\text{rad/s},$$
$$\Omega_2 = 2\pi[0.0023, 0.0275]\,\text{rad/s},$$
$$\Omega_3 = 2\pi[0.0275, 0.0485]\,\text{rad/s},$$
$$\Omega_4 = 2\pi[0.0485, 0.070]\,\text{rad/s}.$$

6. Experimental Results

In this section, some brief considerations about the developed positioning and tracking system are made, and experimental results of its application to real time situations are analyzed.

6.1. Application Description

The positioning and tracking system proposed was implemented in *Matlab*, and can be divided into three main modules: one that addresses the interface with the camera, other that implements the image processing algorithms, and a third related to the estimation process.

1) Interface with the camera: Since the camera used in this work has a discrete and limited range of movements, its desired orientation in each time instant is determined according to a decision system whose aim is to prevent the distance between the image and the target centers from exceeding certain values.

The CCD sensor built-in the camera acquires images with a maximum dimension of 640×480 pixels, which was the resolution chosen for this application. Despite its higher computational requirements, smaller targets can be tracked with an increase on the accuracy of the system.

2) Image processing: Snakes algorithm was implemented using $\alpha = 0.5$ and $\beta = 0.05$, since these were the values that led to the best results.

The noise of the sensor measurements that the image processing module provides to the estimator was characterized according to the procedure described in the end of section 5.1, which led to $\mathbf{R} = diag[3.25^2, 2.04^2, 174.29^2]$.

The application was optimized to follow red targets, whose identification in acquired images is easy, since image segmentation is itself a very complex domain, and does not correspond to the main focus of this work.

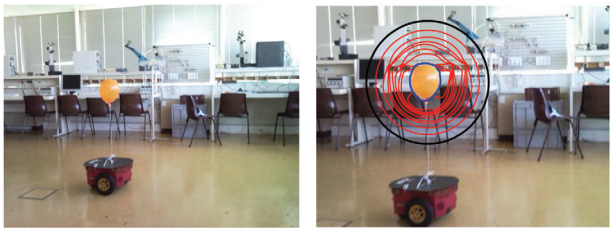
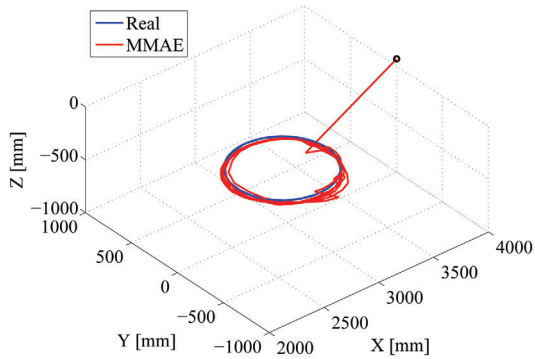
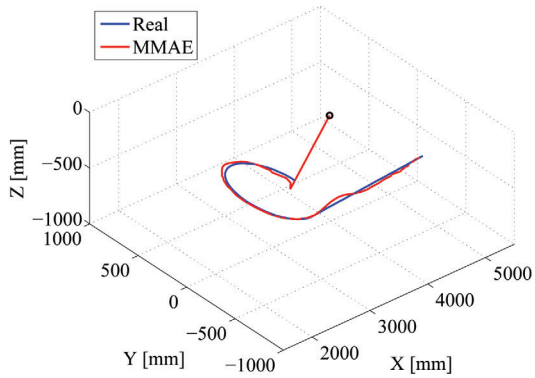


Fig. 4. Real time target tracking. Left: experimental setup; right: target identification, where the initial snake is presented in black, its temporal evolution is presented in red, and the final contour estimate is presented in blue.



(a) Circular trajectory.



(b) Combination of a circular trajectory with a straight line.

Fig. 5. 3D position estimate of a real target. The initial position estimate is presented in black.

3) Estimation process: The adopted MMAE approach was based upon the four initially equiprobable target models proposed in section 5.3, that differ on the target angular velocity values: $2\pi \frac{1}{50}[0, 1, 2, 3]$ rad/s.

Each one of the models requires the knowledge of the power spectral density matrix of the process noise, that is not available. The matrix considered for this quantity was $diag[0.1, 0.1, 0.1]$, since it led to the best experimental results.

Due to limitations imposed by the resources available, the nominal sampling interval for the application was set to 0.5 s. Sometimes, this value may be slightly exceeded. In those situations, the discrete-time system dynamics and the process noise covariance matrix, defined in section 5.1, are corrected accordingly.

6.2. Application Performance

Results presented in this section are relative to tracking a red balloon attached to a robot *Pioneer P3-DX*, as depicted in Fig. 4. This robot was programmed to describe two trajectories, which combine sections with different angular velocities. The target position, linear velocity, and linear acceleration were estimated in real time, and its angular velocity was identified in the two experiments.

In Fig. 5, the 3D nominal and estimated target trajectories are presented. In the first experiment Fig. 6.1, the target describes several times the same circumference with its angular velocity varying over time, and in the second Fig. 5, it describes part of a circumference and then computes to a straight line. The evolution of the target position, velocity, and acceleration along time is depicted in Fig. 6. Despite the significant initial uncertainty in the state estimate, the target position, velocity, and acceleration estimates converge to the vicinity of the real values. However, given the suboptimal nature of the results produced by the Extended Kalman filter in nonlinear applications, in some experimental situations, where an excessively poor initial state estimate was tested, divergence of the filter occurred.

Position, velocity, and acceleration estimation errors are presented in Fig. 7. These quantities have transients in the beginning of the experiments, due to initial state estimation errors, and then converge to a steady state situation with standard deviations below 5.6 cm, 0.9 cm/s, and 0.2 cm/s^2, respectively, in the circular trajectory

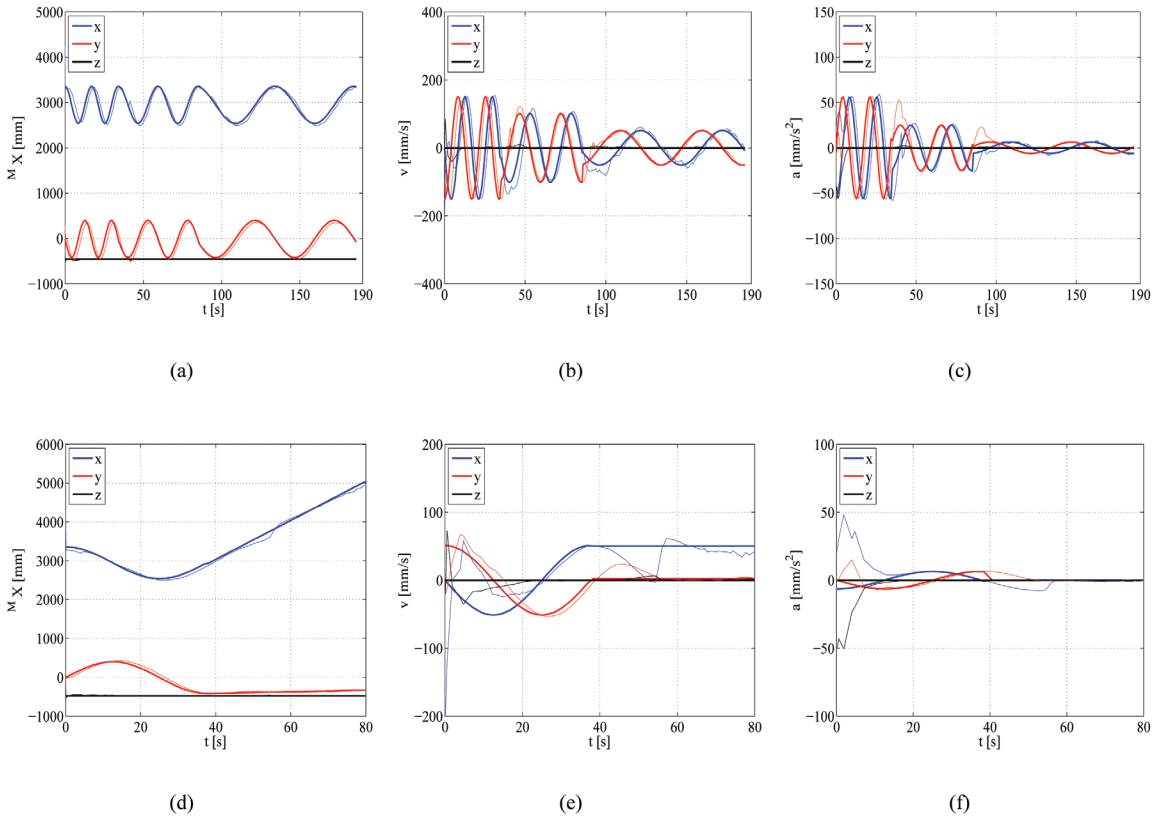


Fig. 6. Position (left pan), velocity (center pan), and acceleration (right pan) estimates of a real target in the world. Figures (a), (b), and (c) correspond to the circular trajectory, and figures (d), (e), and (f) correspond to the combination of a circular trajectory with a straight line. The slender and thicker lines correspond to the estimated and real values, respectively.

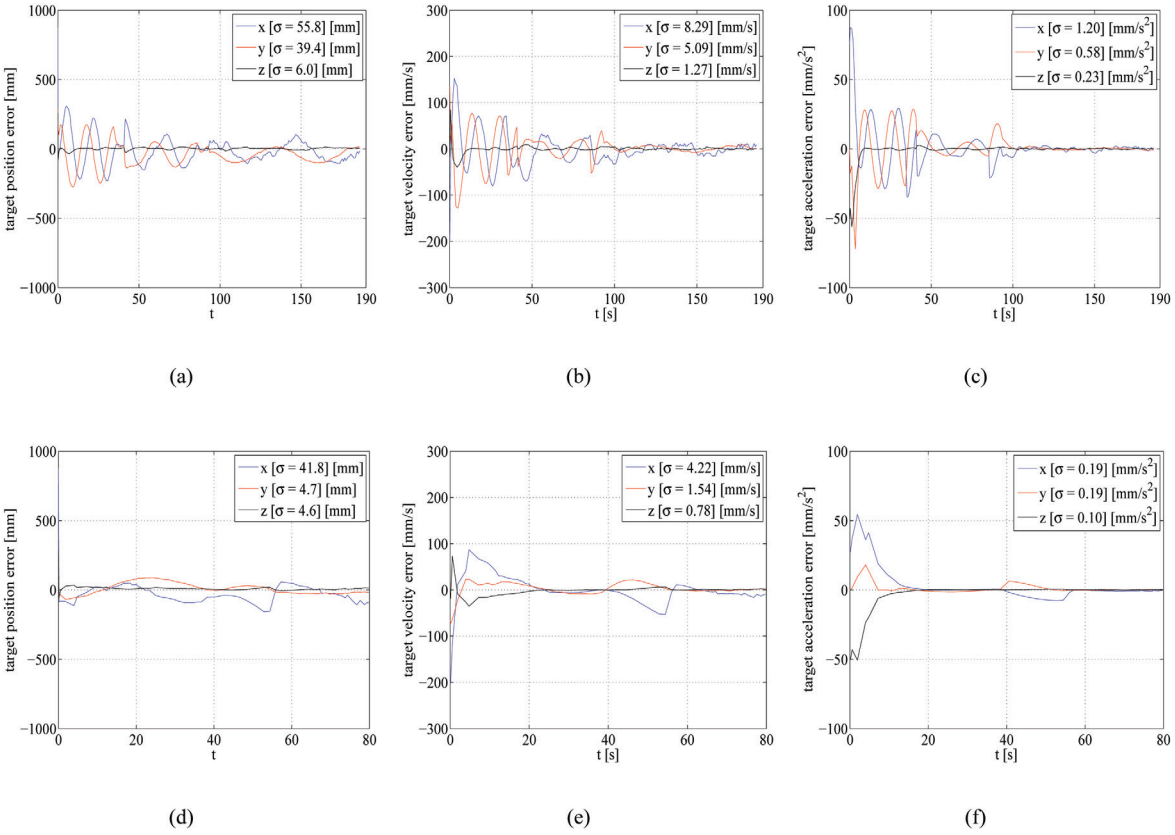


Fig. 7. Position (left pan), velocity (center pan), and acceleration (right pan) estimation errors of a real target in the world. Figures (a), (b), and (c) correspond to the circular trajectory, and figures (d), (e), and (f) correspond to the combination of a circular trajectory with a straight line.

experiment, and below 4.2 cm, 0.5 cm/s, and 0.02 cm/s^2, in the experiment that combines a circular trajectory with the straight line. In all situations depicted, the standard deviation associated with measurements along coordinate x is the largest because this coordinate was approximately in the same direction as the camera optical axis. Therefore, it is intrinsically related to measurements of the depth of the target, which are the noisiest of the three measurements used. There are several reasons that can justify the errors observed: i) the uncertainty associated with the characterization of the real trajectory described by the target; ii) possible mismatches in the models considered for the camera and the target, and iii) incomplete measurement and sensor noise characterization.

Results obtained with the adopted MMAE approach for the two experiments described and for a third experiment are presented in Figs. 8 and 9. As expected, the probability of the model that, in each instant of time, is associated with the angular velocity which is closest to the real target angular velocity, in terms of the BPM, tends to 1. In the circular trajectory experiment, the target describes the same circumference several times with three different angular velocities, which decrease over time. In the beginning of the experiment, the probability associated with the fourth model (the one with the largest angular velocity) converged to 1, and then, a few seconds after the moment when the target angular velocity changed to a value closer to the third model angular velocity, this model was identified as the one that best described the real target. The same behaviour was observed the second time the target changed its angular velocity. As expected, the position, velocity, and acceleration estimates provided by the MMAE degrade slightly when the target changes its angular velocity, Figs. 6 and 7, since the probabilities associated with each model require a certain amount of time to converge for the new values.

A behaviour similar to the one described in the previous paragraph is depicted in Figs. 8(c), 8(d), and 9, which correspond to trajectories more elaborate than the one already addressed: one in which the target moves along a straight line after describing part of a circumference, and other in which it describes two circular trajectories, with different angular velocities, connected by a straight line.

The range of operation of the proposed system depends significantly on the camera used and on the size of the target to be tracked. In the experiments reported, an elliptic

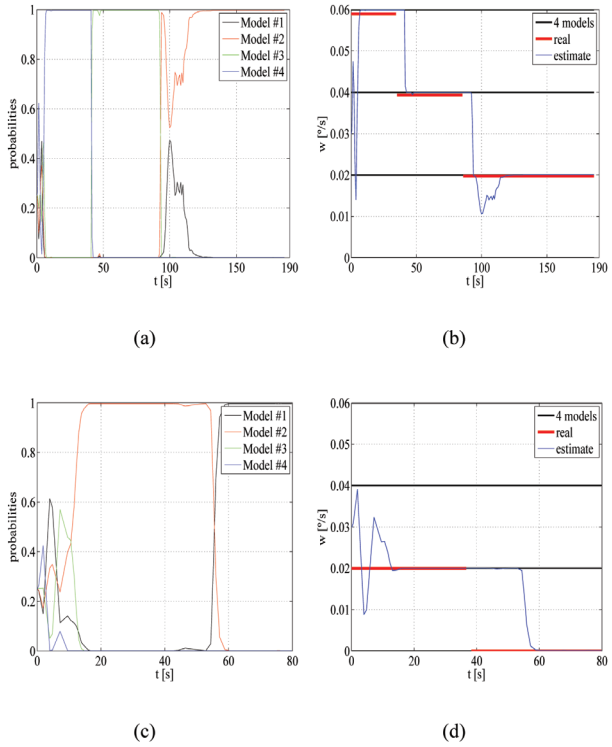


Fig. 8. MMAE evolution over time. Left: *a posteriori* hypothesis probabilities. Right: real and estimated target angular velocities. Figures (a) and (b) correspond to the circular trajectory, and figures (c) and (d) correspond to the experiment that combines a circular trajectory with a straight line.

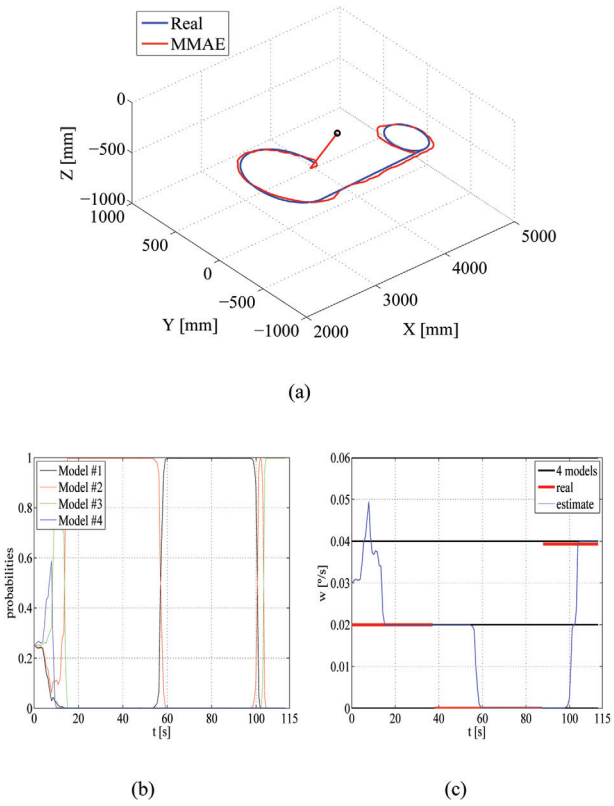


Fig. 9. MMAE evolution over time for the trajectory depicted in (a). Figures (b) and (c) show the *a posteriori* hypothesis probabilities and the real and estimated target angular velocities, respectively.

shape with axes of length 106 mm and 145 mm was identified and located, with the mentioned accuracies up to distances of approximately 7 m from the camera. The lower bound of the range of distances in which the application works properly is limited by the distance at which the target stops being completely visible, filling the camera field of vision. For the target considered, this occurs at distances bellow 40 cm.

7. Conclusions and Future Work

A new indoor positioning and tracking system architecture is presented, supported on suboptimal stochastic multiple-model adaptive estimation techniques. The proposed approach was implemented using a single low cost pan and tilt camera. The real time location of the target, which moves in the 3D real world, was estimated with centimetric accuracy.

The main limitations of the implemented system are the required knowledge on the target dimensions, and the inability to identify targets with colours other than red.

In the near future, an implementation of the developed architecture in **C** will be pursued, which will allow for the tracking of more unpredictable targets. Also, an extension of the proposed architecture to a multiple camera based system is thought. Distances to targets will then be computed resorting to triangulation methods, thus avoiding the requirement on the precise knowledge of their dimensions.

Finally, it is also recommended the integration of an accurate sensor in the vision system that retrieves camera orientation in each time instant, and the implementation of an image segmentation algorithm capable of identifying a wider variety of targets.

References

1. Anderson BDO, Moore JB. Optimal Filtering. Prentice-Hall, USA: Englewood Cliffs, 1979.
2. Bar-Shalom Y, Rong-Li X, Kirubarajan T. Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. John Wiley & Sons, Inc., 2001.
3. Baram Y. Information, consistent estimation and dynamic system identification. PhD thesis, MIT, Cambridge, MA, USA, 1976.
4. Borenstein J, Everett HR, Feng L. Where am I? Sensors and Methods for Mobile Robot Positioning. J. Borenstein, 1996.
5. Caselles V, Catte F, Coll T, Dibos F. A geometric model for active contours. *Numer Math* 1993; 66: 1–31.
6. Devernay F, Faugeras O. Straight lines have to be straight: automatic calibration and removal of distortion from scenes

of structured enviroments. *Mach Vis Appl* 2001; 13: 14–24.
7. Faugeras O, Luong Q. The geometry of multiple images. MIT Press, 2001.
8. Fekri S, Athans M, Pascoal A. Issues, progress and new results in robust adaptive control. Int J Adapt Control Signal Process 2006; 20: 519–579.
9. Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Readings in computer vision: issues, problems, principles, and paradigms, pages 726–740, 1987.
10. Gaspar T, Oliveira P. A single pan and tilt camera architecture for indoor positioning and tracking. In Proceedings of the International Conference on Computer Vision Theory and Applications, volume 1, pages 523–530, Lisbon, Portugal, February 2009.
11. Gaspar T, Oliveira P. Single pan and tilt camera indoor positioning and tracking system. In Proceedings of the European Control Conference, pages 2792–2797, Budapest, Hungary, August 2009.
12. Gelb A. Applied Optimal Estimation. MIT Press, Cambridge, Massachusetts, 2001.
13. Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. *Int J Comput Vis* 1988; 1: 321–331.
14. Kolodziej K, Hjelm J. Local Positioning Systems: LBS Applications and Services. CRC Press, 2006.
15. Lepetit V, Fua P. Monocular model-based 3d tracking of rigid objects: A survey. In Foundations and Trends in Computer Graphics and Vision, pages 1–89, 2005.
16. Leymarie F, Levine MD. Tracking deformable objects in the plane using an active contour model. *IEEE Trans Pattern Anal Mach Intell* 1993; 15: 617–634.
17. Rong Li X, Jilkov VP. Survey of maneuvering target tracking. part i: Dynamic models. *IEEE Trans Aerosp Electron Syst* 2003; 1333–1364.
18. Magill D. Optimal adaptive estimation of sampled stochastic processes. *IEEE Trans Autom Control* 1965; 10: 434–439.
19. McInerney T, Terzopoulos D. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Comput Med Imaging Graph* 1995; 9: 69–83, 1995.
20. Morse AS. Supervisory control of families of linear set-point controllers - part 1: Exact matching. *IEEE Trans Autom Control* 1998; 41: 1413–1431.
21. Terzopoulos D, Fleischer K. Deformable models. *Vis Comput* 1988; 4: 306–331.
22. Thormahlen T, Broszio H, Wassermann I. Robust line-based calibration of lens distortion from a single view. In Proceedings of Mirage, pages 105–112, 2003.
23. Yilmaz A, Javed O, Shah M. Object tracking: A survey. *ACM Comput Surv* 2006; 38: 13.