Single Pan and Tilt Camera Indoor Positioning and Tracking System

Tiago Filipe Pires Gaspar

Abstract— **This work addresses the study, design, analysis, implementation, and validation in real time of an indoor positioning and tracking system. An inexpensive pan and tilt camera based architecture is proposed, where three main modules can be identified: one related to the interface with the camera, supported on 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 following original contributions can be found: i) a new indoor positioning and tracking system architecture; ii) a new lens distortion calibration method, that preserves generic straight lines in images; iii) the use of 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 methods and the resulting architecture, a software package was developed. An accuracy of $20\,cm$ was obtained in a series of indoor experimental tests, for a range of operation of up to ten meter, under realistic conditions.**

Index Terms— **Indoor Positioning and Tracking Systems, Camera Calibration, GVF Snakes, Multiple-Model Adaptive Estimation, Single Camera Vision Systems.**

I. 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 is essential to its navigation and control systems, which justifies the significant effort that has been put into this domain, see [11], [3] 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 the electromagnetic waves, precluding their use. Alternative techniques, such as infrared radiation, ultrasounds, radio frequency, vision has been successfully exploited as reported in detail in [11], and summarized in [7].

The indoor tracking system proposed in this project uses vision technology, since this technique has a growing domain of applicability and allows to achieve acceptable results with very low investment. This system estimates in real time the position, velocity, and acceleration of a target that evolves in an unknown trajectory, in the 3D world, as well as its angular velocity. In order to accomplish this purpose, a new positioning and tracking architecture 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.

This document is organized as follows. In section II the architecture of the developed positioning and tracking system is introduced, as well as the main methodologies and algorithms developed. In section III the camera and lens models are studied in detail. To isolate and identify the target, advanced image processing algorithms are discussed in section IV, and in section V, the used multiple-model nonlinear estimation technique is introduced. In the last two sections, VI and VII, experimental results of the developed system, and concluding remarks and comments on future work, respectively, are presented.

II. System Architecture

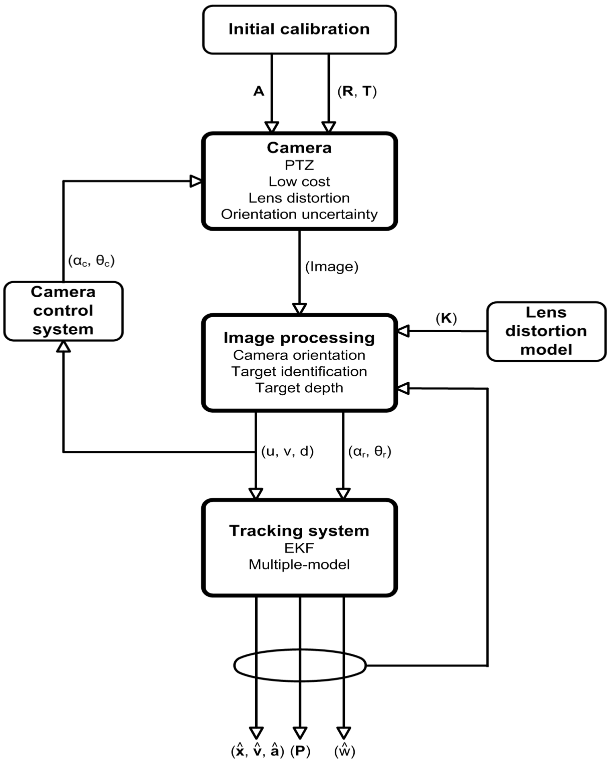

Fig. 1. Tracking system architecture.

In this project a new indoor positioning system architecture is proposed, based on three main modules: one that addresses the interface with the camera, the second that implements the image processing algorithms, and a third responsible for dynamic systems state estimation. The proposed architecture is presented in Fig. 1, and is described next [1].

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 estimate its instantaneous pan and tilt angles (α_r and θ_r, respectively).

The target identification is the main purpose of the image processing block. An active contour method, usually denominated as snakes, was selected to track the important features in the image. The approach selected consists of estimating the target contour, providing the necessary information to compute its center coordinates (u, v) and its distance (d) to the origin of the world reference frame. This quantities correspond to the measurements that are used to estimate the position ($\widehat{\mathbf{x}}$), velocity ($\widehat{\mathbf{v}}$), and acceleration ($\widehat{\mathbf{a}}$) 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 ($\widehat{\mathbf{x}}$, $\widehat{\mathbf{v}}$, and $\widehat{\mathbf{a}}$), to identify the unknown target angular velocity w ($\widehat{w}$), and the estimation error covariance 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 is very simple and consists in computing the pan and tilt angles (α_c and θ_c), that should be sent to the camera at each moment. Large distances between the referred centers are avoided, thus the capability of the overall system to track the targets is increased.

III. Sensor: PTZ camera

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

A. Camera model

1) Pinhole model: given the high complexity of the camera optical system, and the consequent high number of parameters that should be considered in order to model the whole image acquisition process, it is common to explore a linear model to the camera. In this project it was considered the pinhole model [6].

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 corresponding homogeneous coordinates of the same point in the image frame. According to this model, the relation between the coordinates expressed in these two coordinate 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 $\mathbf{P}$ 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, other responsible for projecting 3D points into the image plane, and a third one that changes the origin and units of the coordinate system used to identify each point in the acquired images.

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, $\mathbf{M}_c$ to its correspondent homogeneous coordinates in the camera coordinate frame, $\mathbf{R}$ is a rotation matrix, belonging to SO(3), i.e. verifying $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ 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

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

where (x_c, y_c, z_c) are the cartesian coordinates of a point in the camera coordinate system, (x_p, y_p) are its correspondent coordinates in the image plane, and f is the focal length of the pinhole camera model, where this distance is expressed in mm and is measured between the optical center and the image plane. 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

[1] In this section some quantities are presented informally to augment the legibility of the whole document.

relation between the two referred coordinate systems 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 the conversion factors from mm to pixels. This transformation matrix will be denoted as **A** in this work. It 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 results in the overall expression for the **P** matrix, which is given by $\mathbf{P} = \mathbf{A}.\pi_0.{}^c\mathbf{g}_M$, and establishes the relation between a point in the 3D world and its correspondent in the acquired images.

2) Camera calibration (Method of Faugeras): the use of the previous model implies the determination of the intrinsic and extrinsic parameters referred before. In this work, the classical approach proposed by Faugeras [6] was selected and implemented. This method is linear and implicit, i. e. the intrinsic and extrinsic parameters are computed from the previously estimated **P** matrix.

The disadvantages of this method are the required preparation of the scene in which the camera is inserted, and to disregard the distortion of the lens. However, the impact of these constraints is moderate since the camera in this application is supposed to be placed in a fixed location in the world (the calibration needs to be performed just once). A separate algorithm that compensates for lens distortion is implemented, see section III-C for details. The major advantages are that only one image is required and reliable results can be obtained.

It should also be referred that, despite the ability of the camera selected to move, the calibration process can be performed in a fixed position, leading to the determination of the intrinsic parameters. The extrinsic parameters, that vary according to the camera movements, must be actualized over time (this problem is addressed in the next section).

As stated before, the classical method by Faugeras consists in performing an initial estimation of the projection matrix, that is done from a set of points with known coordinates in world and camera reference frames. Writing (1) and reorganizing the expression obtained to every one of the n points used in the calibration process, and considering that the index i identifies the coordinates of the i^{th} used point, yields, for each point,

$$\begin{bmatrix} x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i z_i & -u_i \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i x_i & -v_i y_i & -v_i z_i & -v_i \end{bmatrix} .\mathbf{p} = 0,$$

with

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{21} & p_{22} & p_{23} & p_{24} & p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}^T,$$

where p_{jk} is the **P** element whose line and column are j and k, respectively.

The previous equations, when applied to the entire set of used points, lead to a system of the form $\mathbf{Lp} = 0$, where **L** is a $2n \times 12$ matrix. The solution of this system corresponds to the eigenvector associated with the smallest eigenvalue of $\mathbf{L}^T\mathbf{L}$, or, equivalently, to the singular vector of **L** associated with the smallest singular value of its SVD (Single Value Decomposition). Since the projection matrix has 12 elements, and each point considered contributes with two equations, there is a minimum of 6 points that must be used in the calibration process.

The resulting **p** vector should be normalized by $\sqrt{p_{31}^2 + p_{32}^2 + p_{33}^2}$, that, as can be concluded from the explicit expression for **P**:

$$\mathbf{P} = \begin{bmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix},$$

corresponds to the norm of the third line vector of the estimated rotation matrix, i.e. $[p_{31}, p_{32}, p_{33}] = [r_{31}, r_{32}, r_{33}]$. This value must be 1 given the orthonormal nature of this family of matrices.

Once estimated the projection matrix, the intrinsic and extrinsic parameters of the camera can be computed as

$$\begin{array}{ccccccc} u_0 & = & \mathbf{p}_1 . \mathbf{p}_3, & & v_0 & = & \mathbf{p}_2 . \mathbf{p}_3, \\ |\alpha_u| & = & ||\mathbf{p}_1 - u_0 \mathbf{p}_3||, & & |\alpha_v| & = & ||\mathbf{p}_2 - v_0 \mathbf{p}_3||, \\ \mathbf{r}_3 & = & \mathbf{p}_3, & & \mathbf{r}_2 & = & \frac{\mathbf{p}_2 - v_0 \mathbf{r}_3}{\alpha_v}, \\ \mathbf{r}_1 & = & \frac{\mathbf{p}_3 - u_0 \mathbf{r}_3}{\alpha_u}, & & t_z & = & p_{34}, \\ t_x & = & \frac{p_{14} - u_0 t_z}{\alpha_u}, & & t_y & = & \frac{p_{24} - v_0 t_z}{\alpha_v}, \end{array}$$

where $\mathbf{p}_k = \begin{bmatrix} p_{k1} & p_{k2} & p_{k3} \end{bmatrix}$, and $\mathbf{p}_i . \mathbf{p}_j$ represents the internal product of the vectors $\mathbf{p}_i$ and $\mathbf{p}_j$.

The signals of α_u and α_v must be chosen according to the relative orientation between the coordinated image axis (in pixels), and the coordinated axis of the image plan (in mm). Moreover, the determinant of a rotation matrix in the special orthogonal group must be 1. If, during the calibration process, a determinant equal to -1 results, the normalization should be done with $-||\mathbf{p}_3||$.

B. PTZ camera internal geometry

The camera used in this project has the ability to describe pan and tilt movements, which makes possible the variation over time of its extrinsic parameters. Thus, the rigorous definition of the rigid body transformation between camera and world reference frames implies the adoption of a model to the camera internal geometry and the study of its direct kinematics.

Since the used *Creative WebCam Live! Motion* camera has a closed architecture, its internal geometry model was estimated from the analysis of its external structure and based on 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 gives the transformation between the frames 0 and 3, and a fifth one between the resulting referential of the previous transformations and the camera reference frame (whose origin coincides with its optical center). Despite the used camera inability to realize roll movements, the inclusion of this degree of freedom in the considered model makes possible to place the camera in any position and with any orientation relatively to world coordinate system.

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.

The previously transformations can be presented as

$$
\begin{aligned}
{}^{M}\mathbf{g}_0 &= \begin{bmatrix} 1 & 0 & 0 & {}^{M}Px_c - \delta \\ 0 & 1 & 0 & {}^{M}Py_c \\ 0 & 0 & 1 & {}^{M}Pz_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^{0}\mathbf{g}_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) & 0 \\ 0 & \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^{1}\mathbf{g}_2 &= \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^{2}\mathbf{g}_3 &= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\theta) & -\cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and} \\
{}^{3}\mathbf{g}_c &= \begin{bmatrix} 0 & 0 & 1 & \delta \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned}
$$

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 the introduced transformations leads to the global transformation between world and camera reference frames:

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

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

The expressions introduced require, however, the knowledge of five parameters: pan, tilt and roll angles, the position of the camera optical center when $\alpha = \theta = \psi = 0^o$ in the world coordinate frame, and the offset between this point and the camera rotation center.

Since there is neither an orientation sensor in the camera nor a closed loop control system capable of placing it in the orientation $\alpha = \theta = \psi = 0^o$, the determination of ${}^{M}\mathbf{P}_c$ was based upon interpolation methods. The approach adopted consisted in calibrating the camera for different orientations relatively 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 based on the utilization of four points with known world coordinates. Calibrating the camera for different orientations, and writing the pair of equations

$$
\begin{aligned}
u\mathbf{P}_3\mathbf{M} - \mathbf{P}_1\mathbf{M} &= 0 \\
v\mathbf{P}_3\mathbf{M} - \mathbf{P}_2\mathbf{M} &= 0
\end{aligned}
$$

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

In what concerns camera orientation, it 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 the output from 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 maintains the value estimated in the calibration process).

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

C. 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 this case and with 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 this distortion by a nonlinear inverse radial distortion function, which corrects measurements in the 2D camera image to those that would have been obtained with a linear pinhole camera model.

The lens distortion compensation method adopted in this project is independent of the calibration process responsible for determining the pinhole model parameters, and is based on the idea that straight lines in the 3D space must remain straight lines in 2D camera images.

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 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

$$r = \sqrt{x^2 + y^2} \quad and \quad r_d = \sqrt{x_d^2 + y_d^2},$$

that results in

$$x = x_d + x_d \sum_{i=0}^{\infty} k_i r_d^{i-1} \quad and \quad y = y_d + y_d \sum_{i=0}^{\infty} k_i r_d^{i-1}. \quad (3)$$

In this project, it were only taken into account the parameters k_3 and k_5, that, as stated in [16] and according to practical tests, are sufficient to obtain good results. Using more parameters brings no major improvement to the approximation of $f(r_d)$ for images in video resolution, and an estimation of less parameters is more robust.

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

$$\left\{ \begin{array}{lcccc} f_{i1} & = & (y_{i1} - \widehat{y}_{i1}(m_i, b_i, x_{i1}))^2 & = & 0 \\ & & \vdots & & \\ f_{iN_p} & = & (y_{iN_p} - \widehat{y}_{iN_p}(m_i, b_i, x_{iN_p}))^2 & = & 0 \end{array} \right. \quad i = 1, \ldots, N_r$$

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 its estimated slope-intercept equation. A set of $N_r * N_p$ nonlinear equations in the parameters to estimate (k_3, k_5, x_0, y_0, m_i, b_i, $i = 1, \ldots, N_r$) results and its solution was found resorting to the Newton's method.

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.

IV. Image Processing

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

A. Target isolation and identification

In this work, the isolation and identification of the target to be tracked in each acquired image is done resorting to an active contours method.

Active contours [10], 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 [10], image segmentation [12], shape modeling [15], [14], or motion tracking [12], as happens in this application.

There are two main types of active contour models: *parametric active contours* [10] and *geometric active contours* [5]. In this project 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. This forces interact with internal ones responsible for holding the curve together and keep it from bending too much.

1) Parametric active contours (traditional method): 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

$$E_{snake} = E_{int} + E_{ext}, \quad (4)$$

that, as can be seen, include 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, snakes evolve in order to minimize the energy functional

$$E = \int_0^1 \frac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] + E_{ext}(\mathbf{x}(s))ds, \quad (5)$$

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. The term $E_{ext}(\mathbf{x}(s))$ corresponds to image contribution to the snake evolution, and is derived from the image so that it has smaller values at the interest points.

There are several typical methods of designing the external energy of an image that would lead a snake towards the desired features [10]. In this project it was considered

$$E_{ext}(x, y) \quad = \quad -|\nabla[G_\sigma(x, y) * I(x, y)]|^2, \quad (6)$$

where $I(x, y)$ represents image intensity at the 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 cause boundaries to become blurry, such values increase the capture range of the active contour.

A curve $\mathbf{x}(s)$ that minimizes (5) must satisfy the Euler equation

$$\alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s) - \nabla E_{ext} = 0, \quad (7)$$

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 (7), 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

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}''(s,t) - \beta\mathbf{x}''''(s,t) - \nabla E_{ext}, \qquad (8)$$

that corresponds to the first member of (7). A solution of (7) 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 minimum that leads the snake towards boundaries other than the desired ones. However, the impact of this limitation can be significantly minimized by the initialization of the snake in the neighborhood of the interest features.

Approximating the derivatives in (8) by the spacial finite differences method, with step h, yields

$$(\mathbf{x}_t)_i = \frac{\alpha}{h^2}(\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}) - \frac{\beta}{h^4}(\mathbf{x}_{i+2} - 4\mathbf{x}_{i+1} + \\ + 6\mathbf{x}_i - 4\mathbf{x}_{i-1} + \mathbf{x}_{i-2}) + \mathbf{F}^{(p)}_{ext}(\mathbf{x}_i), \qquad (9)$$

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

The temporal evolution of the active contour in the image domain occurs according to the expression

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \tau\mathbf{x}^n_t,$$

where τ is the considered temporal step. The iterative process ends when the coordinates of each point of the snake remain approximately constant over time.

In addition to the snakes initialization limitation, it should also be considered the traditional snakes poor convergence to boundary concavities. Therefore, in this project the *gradient vector flow* (GVF) approach proposed by Chenyang Xu and Jerry L. Prince in [17] was followed, where a new class of external forces for active contour models that addresses both problems referred above is introduced.

2) Gradient vector flow snakes: the overall GVF approach proposed in [17] consists in using a new external force, here denoted by $\mathbf{v}(x,y) = \mathbf{F}^{(g)}_{ext}$, which defines the *gradient vector flow* field. Therefore, the new dynamic snake equation is similar to (8), whose potential force $-\nabla E_{ext}$ is replaced with $\mathbf{v}(x,y)$, yielding

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}''(s,t) - \beta\mathbf{x}''''(s,t) + \mathbf{v}. \qquad (10)$$

The parametric curve that solves the above dynamic equation is called *GVF snake*, and is computed numerically by iterative processes, after discretization, in a procedure similar to the one followed in the traditional snake method.

In what concerns GVF field is defined as the vector field $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$ that minimizes the functional

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2|\mathbf{v} - \nabla f|^2 dxdy, \qquad (11)$$

where the indices x and y represent the partial derivatives with respect to x and y, respectively, and f is a scalar field $f(x,y) = -E_{ext}(x,y)$. When $|\nabla f|$ is small, the energy is dominated by the sum of squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $\mathbf{v} = \nabla f$. This keeps $\mathbf{v}$ nearly equal to the gradient of f when it is large (making this approach similar to the traditional one in this regions), but forces the field to be slowly-varying in regions where $I(x,y)$ is approximately constant. This second feature contributes to extend the capture range of the traditional external force fields, since the small magnitude of $|\nabla f|$ in regions where $I(x,y)$ is approximately constant does not contribute to pull snakes towards boundaries.

The parameter μ is a regularization parameter governing the tradeoff between the first and the second terms in the integrand. This parameter should be set according to the amount of noise present in the image, i.e. images with more noise require the choice of larger values for the parameter μ.

Using the calculus of variations [9], it can be shown that the GVF field that minimizes (11) can be found by solving the following Euler equations:

$$\begin{aligned} \mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) &= 0 \qquad (12)\\ \mu\nabla^2 v - (v - f_y)(f_x^2 + f_y^2) &= 0, \qquad (13) \end{aligned}$$

where ∇^2 is the Laplacian operator.

The solution of this equations can be computed by means of an iterative numerical procedure, that, as deduced in [17], corresponds to propagate the GVF field components according to the iterative expressions

$$u^{n+1}_{i,j} = (1 - b_{i,j}\Delta t)u^n_{i,j} + r(u^n_{i+1,j} + u^n_{i,j+1} + \\ + u^n_{i-1,j} + u^n_{i,j-1} - 4u^n_{i,j}) + c^1_{i,j}\Delta t \qquad (14)$$

$$v^{n+1}_{i,j} = (1 - b_{i,j}\Delta t)v^n_{i,j} + r(v^n_{i+1,j} + v^n_{i,j+1} + \\ + v^n_{i-1,j} + v^n_{i,j-1} - 4v^n_{i,j}) + c^2_{i,j}\Delta t, \qquad (15)$$

where

$$\begin{aligned} b(x,y) &= f_x(x,y)^2 + f_y(x,y)^2, \qquad (16)\\ c^1(x,y) &= b(x,y)f_x(x,y), \text{ and} \qquad (17)\\ c^2(x,y) &= b(x,y)f_y(x,y). \qquad (18) \end{aligned}$$

The notation adopted is the one proposed by Xu and Prince in [17], so that f_x and f_y correspond to the partial derivatives of f with respect to x and y; indices i, j and n correspond to x, y and t, respectively; Δt corresponds to the time step for each iteration, and

$$r = \frac{\mu\Delta t}{h^2}, \qquad (19)$$

with $h = \Delta x = \Delta y$ (Δx and Δy correspond to the spacing between pixels).

According to numerical analysis theory [1], stability of equations (14) and (15) is guaranteed whenever the restriction

$$0 < \triangle t \leq \frac{h^2}{4\mu + h^2||b||}, \quad ||b|| = \max_{\forall_{i,j}} b_{i,j},$$

is verified. As can be concluded from the expressions above, convergence of the iterative process can be made faster on coarser images, i.e. for larger values of the spatial sampling h. On the other hand, smoother GVF fields, with larger values of the parameter μ, make the convergence rate slower. These last cases correspond to smaller values of the sampling period $\triangle t$.

B. 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. Its real dimensions in the 3D world, and the knowledge of the camera intrinsic and extrinsic parameters, allows to establish metric relations between image and world quantities. Estimates on the depth of the target can then be obtained. A complete stochastic characterization can be found in [7] and will be the measurements of the estimation method detailed next.

The use of triangulation methods for at least two cameras, would allow the computation of the target distance without further knowledge on the target. However, the present tracking system uses a single camera. Thus, additional information must be available. In this work, it is assumed that the target dimensions are known.

V. Tracking System

In this section, the implemented nonlinear estimation methods is described. Estimates on the target position, velocity and acceleration, in the 3D world, are provided and 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.

A. Extended Kalman filter

The Kalman filter [8] 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 nod 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) [8], and have the advantage of being computationally efficient, which is essential in a real time applications.

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

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \tag{20}$$

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

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \tag{21}$$

where the index k represents time, 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 independent, i.e. $E[\mathbf{w}_k {\mathbf{v}_k}^T] = 0$ and with Gaussian probability density functions

$$\begin{aligned} p(\mathbf{w}_k) &\sim N(0, \mathbf{Q}_k) \\ p(\mathbf{v}_k) &\sim N(0, \mathbf{R}_k), \end{aligned}$$

where $\mathbf{Q}_k$ and $\mathbf{R}_k$ represent the process and measurement noise covariance matrices, respectively.

Introducing the notation:

- $\widehat{\mathbf{x}}_k^-$: *a priori* state estimate at step k given the estimated state at step $k-1$;
- $\widehat{\mathbf{x}}_k$: *a posteriori* state estimate at step k (given measurement $\mathbf{z}_k$);
- $\mathbf{e}_k^- = \mathbf{x}_k - \widehat{\mathbf{x}}_k^-$: *a priori* estimate error;
- $\mathbf{e}_k = \mathbf{x}_k - \widehat{\mathbf{x}}_k$: *a posteriori* estimate error;
- ${\mathbf{P}_k}^-$: *a priori* estimate error covariance;
- $\mathbf{P}_k$: *a posteriori* estimate error covariance;
- $\mathbf{A}_k$: Jacobian matrix of partial derivatives of f with respect to $\mathbf{x}$;
- $\mathbf{W}_k$: Jacobian matrix of partial derivatives of f with respect to $\mathbf{w}$;
- $\mathbf{H}_k$: Jacobian matrix of partial derivatives of h with respect to $\mathbf{x}$;
- $\mathbf{V}_k$: Jacobian matrix of partial derivatives of h with respect to $\mathbf{v}$;

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 [8] and referenced therein:

Predict step

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

Update step

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

where $\mathbf{K}_k$ is the Kalman gain.

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 [13], was selected.

According to the adopted model and considering $\mathbf{x} = [\mathsf{x}, \dot{\mathsf{x}}, \ddot{\mathsf{x}}, \mathsf{y}, \dot{\mathsf{y}}, \ddot{\mathsf{y}}, \mathsf{z}, \dot{\mathsf{z}}, \ddot{\mathsf{z}}]^T$ the state of the target, yields

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &= diag[\mathbf{F}(w), \mathbf{F}(w), \mathbf{F}(w)]\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{F}(w) &= \begin{bmatrix} 1 & \frac{sin(wT)}{w} & \frac{1-cos(wT)}{w^2} \\ 0 & cos(wT) & \frac{sin(wT)}{w} \\ 0 & -wsin(wT) & cos(wT) \end{bmatrix},\end{aligned} \tag{22}$$

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 motion, 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 the matrix $\mathbf{W}$, that, as follows directly form its definition and (22), corresponds to the identity matrix of dimensions 9×9.

Note that the presented dynamics and the process noise covariance matrices depend explicitly upon the target angular velocity. Since this information is not available, its identification was done by means of a multiple model adaptive estimation approach (MMAE), that will be studied ahead. An alternative would be to add w to the state vector, however this option would increase its dimension and would make the model, that is linear in the state, highly nonlinear.

In what concerns the sensor measurements available in each time instant, that correspond to the target center coordinates (u, v) and target distance (d) to the origin of world reference frame, are given by

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

where p_{ij} is the projection matrix element in the 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 (23), function h, defined in (21), is nonlinear, which justifies the implementation of the EKF. The linearization of this function leads to the matrix $\mathbf{H}$ (not included in this document, due to space constraints, see [7] for details) and to the matrix $\mathbf{V}$, that corresponds to the identity matrix of dimensions 3×3, as follows directly form its definition and (23).

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

B. Multiple-model

The model considered to the target requires the knowledge of its angular velocity. However, this value is not known in real applications, which led us to the application of a multiple model based approach, identifying simultaneously some parameters of the system and estimating its state.

The implemented method, known as *Multiple-Model Adaptive Estimation (MMAE)* [2], considers several models to a system that differ in a parameters set (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 a model i evolves over time from an initial estimate (p_0^i), according to the following expression (see [2] for details):

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

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 N corresponds to the number of considered models, $i = 1, \ldots, N$ to each one of these models, $\mathbf{r}^i$ to the residual vector of the i^{th} Kalman filter (difference between the sensor measurements and the ones predicted by the model i), $\mathbf{S}^i$ to the residual covariance matrix associated with the i^{th} Kalman filter, m to the number of measurements (number of elements of $\mathbf{z}_k$), and k to 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

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

It should be stressed that the methods used to compute the *a posteriori* probabilities of each model and the final state estimate are optimal if each one of the individual estimates is optimal. However, this is not the case in this application, since the known solutions to nonlinear estimation problems at present do not provide optimal results.

VI. Experimental Results

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

A. Application description

The positioning and tracking system proposed in this project 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 project has a discrete and limited range of movements, its orientation in each time instant is determined according to a decision system whose aim is to avoid that the distance between the image and the target centers exceed certain values.

The CCD sensor built-in the camera acquires images with a maximum dimension of $640 \times 480\,pixels$, which is 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: the GVF method studied before is an iterative process that proved to be too slow for a real time application. The computation of the GVF field halves easily the number of images processed by unit of time, and, as a result, the sampling interval doubles. Given the significant impact of this parameter in the performance of the system, this algorithm was not included in the developed application, that resorts uniquely to the traditional snake method. The used values of α and β were 0.5 and 0.05, respectively, since these values were the ones that led to better results.

The developed application is 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 thesis.

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

Each one of the referred models requires the knowledge of the power spectral density matrix of the process noise, that is not available. The matrix considered to this quantity was $diag[0.1, 0.1, 0.1]$, since it led to the best experimental results. The sampling interval of the developed application was made variable, however the use of the previously referred parameters imposed an inferior bound of approximately $0.5\,s$.

B. Application performance

The results presented in this section are relative to the tracking of a red balloon attached to a robot *Pioneer P3-DX*, as depicted in Fig. 2, programmed to describe a circular trajectory, namely its position, velocity, acceleration, and angular velocity.

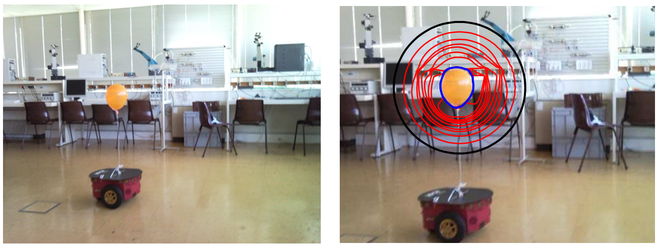

Fig. 2. Real time target tracking. Left: Experimental setup; Right: Target identification, where the initial snake is presented in black, its temporal evolution is presented in red, and the contour final estimate is presented in blue.

In Fig. 3, the 3D nominal and estimated target trajectories are presented.The target position, velocity and acceleration along time are depicted in Fig. 4. Despite the significant initial uncertainty in the state estimate, the target position, velocity, and acceleration estimates converge to the vicinity of the real values. Moreover, given the suboptimal nature of the results produced by the extended Kalman filter in nonlinear applications, in some experimental cases where an excessively poor initial state estimate was tested, divergence of the filter occurred.

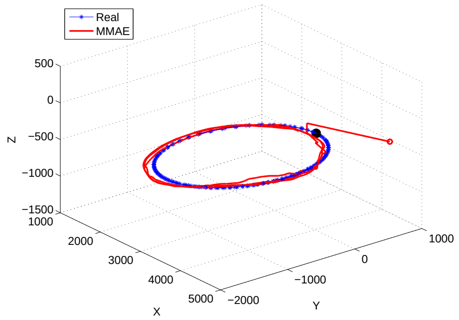

Fig. 3. 3D position estimate of a real target. The real position of the target in the initial instant is presented in black.

The position, velocity, and acceleration estimation errors are presented in Fig. 5. These quantities have large transients in the beginning of the experiment, due to the initial state estimation error, and decrease quickly to values beneath $20\,cm$, $4\,cm/s$, and $0.5\,cm/s^2$, respectively. There are several reasons that can justify the errors observed: i) the uncertainty associated with the characterization of the real trajectory described by the

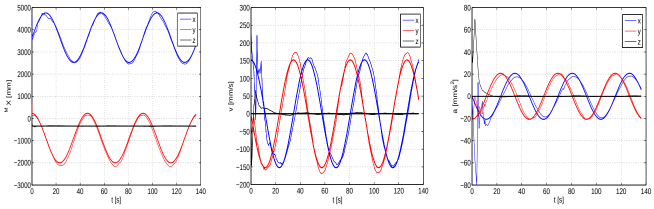

Fig. 4. Position (left pan), velocity (center pan), and acceleration (right pan) estimates of a real target in the world. The slender and tickler lines correspond to the estimated and real values, respectively.

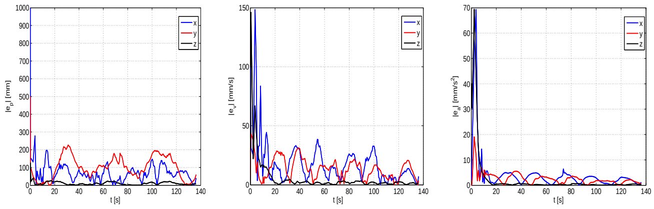

Fig. 5. Position (left pan), velocity (center pan), and acceleration (right pan) estimation error of a real target in the world.

target, and ii) possible mismatches between the models considered for the camera and target, and iii) incorrect measurement and sensor noise characterization.

The results of the adopted MMAE approach are presented in Fig. 6. For the trajectory reported the real target angular velocity is $2\pi 0.0217\, rad/s$. Thus, the probability associated to the model closer to the real target tends to 1 along the experiment, as depicted on the left panel of Fig. 6. On the right panel of that figure, the real and estimated angular velocities are plotted.

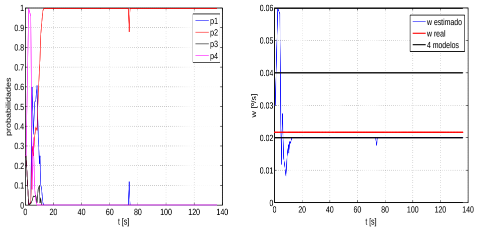

Fig. 6. MMAE evolution over time. On the left, the *a posteriori* hypothesis probabilities. On the right, real (red) and estimated (blue) target angular velocity .

In what concerns the range of operation for the proposed system, it depends significantly on the camera used and on the size of the target to be tracked. In the experiments reported, an elliptic 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 $40\, cm$.

VII. 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, estimating the real time location of a target which moves in the 3D real world with accuracies on the order of $20\, cm$.

The main limitations of the implemented system are the required knowledge on the target dimensions, and with the inability to identify targets with colors 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 a sensor in the vision system that retrieves camera orientation in each time instant, and the implementation of an image segmentation algorithm that can identify a wider variety of targets.

References

[1] W. F. Ames. *Numerical Methods for Partial Differential Equations*. New York: Academic, 3rd ed. edition, 1992.

[2] M. Athans and C.B. Chang. *Adaptive Estimation and Parameter Identification using Multiple Model Estimation Algorithm*. MIT Lincoln Lab., Lexington, Mass., June 1976.

[3] Y. Bar-Shalom, X. Rong-Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, Inc., 2001.

[4] J. Borenstein, H. R. Everett, and L. Feng. *Where am I? Sensors and Methods for Mobile Robot Positioning*. Editado e compilado por J. Borenstein, 1996.

[5] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numer. Math.*, 66:1–31, 1993.

[6] O. Faugeras and Q. Luong. *The geometry of multiples images*. MIT Press, 2001.

[7] T. Gaspar. Sistemas de seguimento para aplicações no interior. Master's thesis, Instituto Superior Técnico, 2008.

[8] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts, 2001.

[9] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Dover Publ., 2000.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vis.*, 1:321–331, 1987.

[11] K. Kolodziej and J. Hjelm. *Local Positioning Systems: LBS Applications and Services*. CRC Press, 2006.

[12] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:617–634, 1993.

[13] X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. part i: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1333–1364, 2003.

[14] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Comput. Med. Imag. Graph*, 9:69–83, 1995.

[15] D. Terzopoulos and K. Fleischer. Deformable models. *Vis. Comput.*, 4:306–331, 1988.

[16] T. Thormahlen, H. Broszio, and I. Wassermann. Robust line-based calibration of lens distortion from a single view. *Mirage 2003*, pages 105–112, 2003.

[17] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Processing*, 7:359–269, 1998.