# FUZZY VISUAL CONTROL FOR MEMORY-BASED NAVIGATION USING THE TRIFOCAL TENSOR

**HÉCTOR M. BECERRA**
*Centro de Investigación en Matemáticas (CIMAT)*
*Jalisco S/N, Col. Valenciana, C.P. 36240, Guanajuato, Gto., Mexico.*
*E-mail: hector.becerra@cimat.mx*

**ABSTRACT**— In this paper, a fuzzy control scheme for visual path-following of wheeled mobile robots is presented. It relies on the feedback of a geometric constraint: the trifocal tensor (TT). The TT is computed from the image currently seen by an onboard camera and the sequence of target images previously acquired, which defines the visual path. Only one element of the TT is needed for feedback, which provides information of the robot's deviation from the path. This is used in a set of Mamdani-type fuzzy rules that mimics the human action of driving a vehicle. The translational velocity is also adapted by a fuzzy system in function of the path's curvature, inferred from the TT computed a priori using the set of target images. The use of fuzzy control allows achieving an effective and simple controller that does not need a time-varying reference to be tracked while the resultant robot velocities are smooth or piece-wise constant. The validity and performance of the approach is shown through simulations using synthetic images.

**Key Words:** Visual navigation, fuzzy visual control, visual path-following, visual memory, mobile robots.

## 1. INTRODUCTION

The locomotion of most of service robots is based on a wheeled platform, so that, the strategies to improve their navigation capabilities result of great interest for the robotics research community. In this sense, machine vision has shown good advantages as sensor for robot control (visual servoing) [1] and navigation [2]. In this paper, a control scheme for following a visual path is presented. The visual path is extracted from a *visual memory*. The proposed scheme uses the trifocal tensor (TT) as feedback information.

The concept of visual memory means that there is a learning stage in which a set of target images are stored. These key images define the path to be replayed in an autonomous stage. This strategy has been introduced for omnidirectional images in [3]. Recently, there are contributions toward the development of autonomous vehicles under this approach. Some of them are position-based approaches, in which, a 3D reconstruction is carried out either using an EKF-based SLAM [4] or a structure from motion algorithm through bundle adjustment [5]. A complete map building is avoided in [6] by relaxing to a local Euclidean reconstruction from the essential matrix. In visual control, image-based approaches generally offer a faster closed loop control with good performance. An approach that uses the centroid of the abscissas of the feature points is presented in [7]. Most of the mentioned approaches suffer the problem of generating discontinuous rotational velocities when a new key image must be reached. This problem is tackled in [8] for conventional cameras, where the authors propose a varying reference instead of a constant one. The epipolar geometry has been used as feedback information for mobile robot navigation [9]; however, this geometric constraint is less robust than the TT and it presents the problem of short baseline.

Fuzzy logic control has been extensively used for the control of nonlinear systems by means of two main approaches: the classical Mamdani-type control, e.g. [10], and the Takagi-Sugeno type control, e.g. [11]. Many works have exploited fuzzy logic for positioning control of nonholonomic mobile robots, e.g. [12]. The combination of fuzzy logic and computer vision has been successfully addressed for different problems,

as in the problem of vision-based human motion analysis [13]. Particularly, fuzzy logic has been introduced for visual servoing of robot manipulators in some works, e.g. [14]. Fuzzy control has been extended for visual control of wheeled mobile robots in works such as [10, 15]. In those works, the approach relies on the following of a lane painted on the floor, which is extracted and used as setpoint to steer the robot accordingly. The visual memory approach is a natural way to model an environment avoiding the use of artificial landmarks.

In this paper, Mamdani-type fuzzy controllers are proposed to solve the visual path-following problem using a sequence of key images. Due to that the problem mainly consists in computing an adequate rotational velocity to follow the path, a set of Mamdani-type fuzzy rules are an efficient way to infer the velocity value. To the author's knowledge, the benefits of fuzzy logic are exploited for the first time in this paper in the framework of visual navigation based on a visual memory. A previous work proposed a qualitative visual navigation scheme that is based on some deterministic rules instead of fuzzy rules [16]. Thus, a strong motivation of this work was to evaluate a more natural and more flexible ruled-based approach than the existing deterministic one. Because of the knowledge of the author on the problem of visual path-following obtained from the previous work [17], it was feasible to address this problem by designing a fuzzy control. The main motivation of the work herein was to achieve a simpler controller than the one presented in [17] by avoiding the use of a time-varying reference to be tracked while the computed velocities remains smooth or piece-wise constant.

The control scheme proposed in this paper uses the value of one element of the trifocal tensor as the only required feedback information. Thus, this approach does not require explicit pose parameters estimation unlike position-based schemes [4, 5]. In this work, the visual servoing problem is transformed in a reference tracking problem for the selected tensor element. It avoids the recurrent problem of discontinuous rotational velocity at key image switching of image-based schemes [6, 7, 16]. The use of the TT allows the gathering of many visual features into a single measurement, so that, undesired pseudoinverse of matrices is not needed. The TT as visual measurement improves the robustness of the control scheme against image noise [8] and it avoids the problem of short baseline [9]. As used in this work, the TT gives the benefit of taking into account valuable a priori information that is available in the visual memory and that is not exploited in previous image-based approaches. This information is used to adapt the translational velocity and also to achieve piece-wise constant rotational velocity according to the learned path. Additionally, the proposed scheme can be applied not only to conventional cameras but also to any vision systems having approximately a unique center of projection, which increase the applicability of the scheme.

The paper is organized as follows. Section 2 introduces the robot and camera model, and the TT for generic cameras. Section 3 describes the control strategy exploiting the TT. Section 4 introduces a nonlinear controller as preamble to the fuzzy control scheme detailed in Section 5. Section 6 presents the performance of the visual navigation via realistic simulations using synthetic images and Section 7 concludes the work.

## 2. MATHEMATICAL MODELING

### *2.1 Robot Kinematics*

Let $\chi = (x, y, \phi)^T$ be the state vector of a differential drive robot (Fig. 1(a)), where $x(t)$ and $y(t)$ are the robot position coordinates in the plane, and $\phi(t)$ is the robot orientation. The kinematic model of the robot expressed in state space can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{1}$$

being $v(t)$ and $\omega(t)$ the translational and rotational input velocities, respectively.

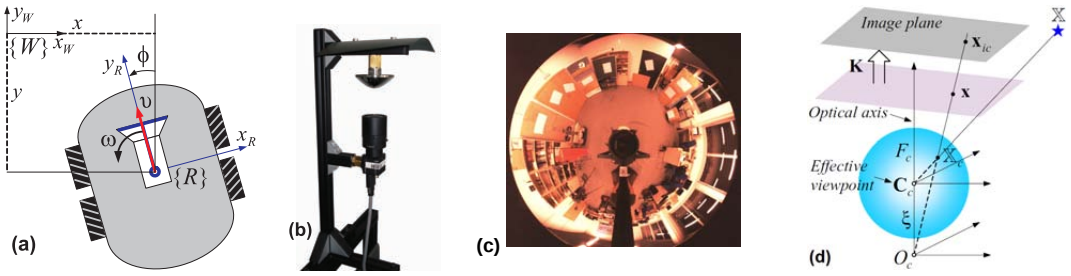


**Figure 1: Representation of the robot model and the camera model. (a) Robot frame definition. (b) Example of central catadioptric vision system. (c) Example of an image captured by a catadioptric system. (d) Generic camera model of central cameras.**

## *2.2 The Trifocal Tensor (TT) for Generic Cameras*

The TT encapsulates all the geometric relations between three views independently of the structure of the scene and depending nonlinearly on the motion parameters among the three views. The generic camera model [18] allows the computation of a geometric constraint, like the TT, in the same way for any central vision system, i.e., any imaging system having approximately a single center of projection. This model allows us to represent from conventional cameras to catadioptric cameras like the one shown in Fig. 1(b-c).

The unified projection model [18] describes the image formation as a composition of two central projections. The first is a central projection of a 3D point onto a virtual unitary sphere and the second is a perspective projection onto the image plane through a collineation $\mathbf{K}$. Let denote a 3D point as $\mathbb{X}$, and its corresponding coordinates as $\mathbf{X}$. Thus, point coordinates on the sphere $\mathbf{X}_c$ can be computed from point coordinates on the normalized image plane $\mathbf{x}$ (refer to Fig. 1(d)) and the sensor parameter $\xi$ as follows:

$$\mathbf{X}_c = \left(\eta^{-1} + \xi\right) \bar{\mathbf{x}}, \text{ with } \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^T & \frac{1}{1+\xi\eta} \end{bmatrix}^T, \tag{2}$$

where $\eta = \frac{-\gamma - \xi\left(x^2+y^2\right)}{\xi^2(x^2+y^2)-1}$ and $\gamma = \sqrt{1+(1-\xi^2)\left(x^2+y^2\right)}$. In this work, the camera is assumed to be calibrated, which allows us to exploit the representation of the points on the unitary sphere.

The TT has 27 elements and it can be expressed by three 3×3 matrices ($\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$). There are 26 independent ratios apart from the common overall scaling of the matrices. In this work, image points are used as visual features. Consider three corresponding points projected on the unitary sphere $\mathbf{p}$, $\mathbf{p}'$ and $\mathbf{p}''$ and expressed in homogeneous coordinates. The incidence relation between these points is given by:

$$[\mathbf{p}']_\times \left(\sum_i p^i \mathbf{T}_i\right) [\mathbf{p}'']_\times = \mathbf{0}_{3\times 3},$$

where $[\mathbf{p}]_\times$ is the common skew symmetric matrix. This expression provides a set of nine equations, however, only four of them are linearly independent. Thus, seven triplets of point correspondences are needed to compute the 27 elements of the tensor by solving a SVD problem for the set of linear equations.

In the case in which the three cameras are located in the same plane, for instance, with the same vertical position from the ground, several elements of the tensor are zero and only 12 elements are in general non-null. Fig. 2 depicts the upper view of three cameras with global reference frame in the third view, in such a way that the corresponding locations are $\mathbf{C}_1 = (x_1, y_1, \phi_1)$, $\mathbf{C}_2 = (x_2, y_2, \phi_2)$ and $\mathbf{C}_3 = (0, 0, 0)$. Analytically, the TT can be deduced for this framework as done in [19], resulting in that the non-null elements are given as:

$$\begin{aligned}
T^m_{111} &= -t_{x_1}\cos\phi_2 + t_{x_2}\cos\phi_1,\ T^m_{113} = t_{x_1}\sin\phi_2 + t_{y_2}\cos\phi_1,\\
T^m_{131} &= -t_{y_1}\cos\phi_2 - t_{x_2}\sin\phi_1,\ T^m_{133} = t_{y_1}\sin\phi_2 - t_{y_2}\sin\phi_1,\\
T^m_{212} &= -t_{x_1},\ T^m_{221} = t_{x_2},\ T^m_{223} = t_{y_2},\ T^m_{232} = -t_{y_1},\\
T^m_{311} &= -t_{x_1}\sin\phi_2 + t_{x_2}\sin\phi_1,\ T^m_{313} = -t_{x_1}\cos\phi_2 + t_{y_2}\sin\phi_1,\\
T^m_{331} &= -t_{y_1}\sin\phi_2 + t_{x_2}\cos\phi_1,\ T^m_{333} = -t_{y_1}\cos\phi_2 + t_{y_2}\cos\phi_1,
\end{aligned} \tag{3}$$

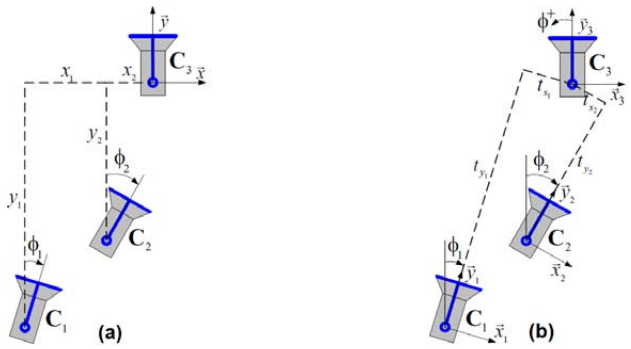


**Figure 2: Geometry between three camera locations in the plane. (a) Absolute locations with respect to a reference frame in $\mathbf{C}_3$. (b) Relative locations.**

where $t_{x_i} = -x_i \cos\phi_i - y_i \sin\phi_i$, $t_{y_i} = x_i \sin\phi_i - y_i \cos\phi_i$ for $i = 1, 2$ and the superscript $m$ states that they are the tensor elements given by metric information. In practice, the estimated tensor has an unknown scale factor and this factor changes as the robot moves. A common scale can be fixed during the navigation by normalizing each element of the tensor using $T_{ijk} = T^e_{ijk}/T_N$, where $T^e_{ijk}$ are the estimated TT elements obtained from point matches, $T_{ijk}$ are the normalized elements and $T_N$ is a suitable normalizing factor. It can be seen from (3) that $T_{212}$ and $T_{232}$ are constant and non-null, assuming that the camera location $\mathbf{C}_1$ is different to $\mathbf{C}_3$. Therefore, any of these elements is good option as normalizing factor.

## 3. CONTROL STRATEGY FOR MEMORY-BASED NAVIGATION

This section describes the proposed control strategy for wheeled mobile robot navigation based on the visual memory approach. First, such approach is briefly described. Next, the way in which the TT is used in this approach is presented.

### *3.1 The Visual Memory Approach*

The navigation based on a visual memory consists of two stages. The first one is a learning stage where the visual memory is built. In this stage, the user guides the robot along the environment where it is allowed to move. A sequence of images are stored from the onboard camera during this stage in order to get a representation of the environment. It is assumed that during learning, the translational velocity is never zero. From all the captured images, a reduced set is selected as key images by ensuring a minimum number of shared point features between two images. Thus, the visual memory defines a path to be replayed in the autonomous navigation stage. It is assumed that $n$ key images are chosen and that these images are separated along the path in the Cartesian space by a minimum distance $d_{\min}$. For more details about the visual memory building and key images selection refer to [6].

### *3.2 The TT for Memory-Based Navigation*

The TT has been exploited for the positioning of a mobile robot in [19] and [20]. In these works, both, the rotational and the translational velocities are computed from the elements of the tensor, which are driven to zero in order to accomplish the positioning task. The visual path-following problem only requires a rotational velocity to correct the deviation from the desired path. Consider two images $I_1(\mathbf{K}, \mathbf{C}_1)$ and $I_3(\mathbf{K}, \mathbf{C}_3)$ belonging to the visual path and the current view of the onboard camera $I_2(\mathbf{K}, \mathbf{C}_2)$. As can be seen in Fig. 3(a), the element $T_{221}$ of the TT (3) provides direct information of the lateral deviation of the

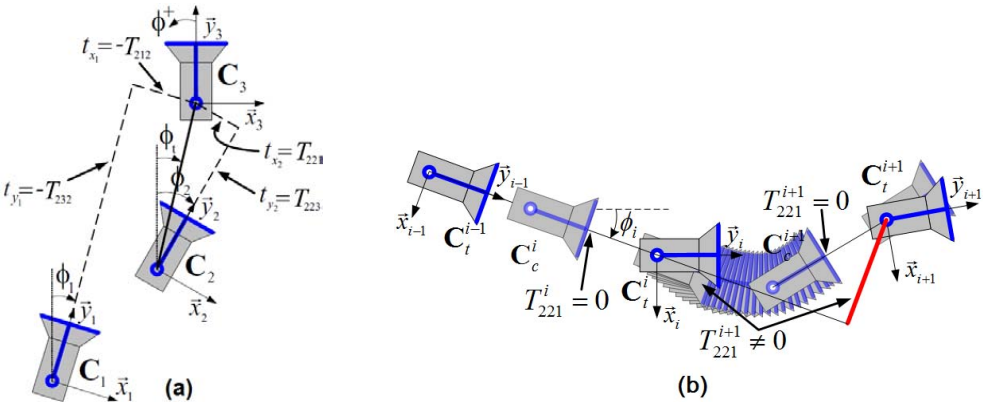


**Figure 3: Memory-based navigation using the TT. (a) Relative locations between cameras up to a scale provided by the TT. (b) Control strategy based on driving to zero the element of the trifocal tensor $T_{221}$.**

current location $\mathbf{C}_2$ with respect to the target $\mathbf{C}_3$. It is worth emphasizing that the 1D TT does not provide this particular information of lateral error [20], so that, the 2D TT is chosen.

Assuming that the center of projection coincides with the rotational axis of the robot, the element $T_{221}$ of the tensor is related to the current location of the robot as follows:

$$T^m_{221} = t_{x_2} = -x_2 \cos\phi_2 - y_2 \sin\phi_2.$$

It can be seen that if $T^m_{221} = 0$ then $\phi_2 = \phi_t = -\tan(x_2/y_2)$, and consequently the current camera $\mathbf{C}_2$ is looking directly toward the target. Thus, it is proposed to compute the rotational velocity from feedback information given by the element $T_{221}$. The control goal is to drive this element with smooth evolution from its initial value to zero before reaching the next key image of the visual path. A reference tracking control problem can be defined in order to avoid discontinuous rotational velocity in the switching of key images. It is also possible to exploit the a priori information provided by the visual path to compute an adequate translational velocity and a nominal rotational velocity according to the shape of the path.

## 4. NONLINEAR CONTROLLER FOR REFERENCE TRACKING

In this section, a control law that corrects the lateral deviation of the robot with respect to the learned visual path for each key image is described. As indicated in Fig. 3(a) the tensor element $T_{221}$ corresponds to the relative lateral position between the current location $\mathbf{C}_2$ and the target location $\mathbf{C}_3$. As shown in Fig. 3(b), if $T_{221}$ is zero, the camera at the current location is pointing directly toward the target location. Then, the rotational velocity control must take to zero the tensor element $T_{221}$, in such a way that this action drives the robot to point toward the next target while remains moving forward. It is desirable that the evolution of $T_{221}$ will be smooth for each segment of the visual path to avoid discontinuities in the rotational velocity. To achieve that, the fed back error has to be zero at each key image switching, so that a null rotational velocity is obtained at switching instants and discontinuities are avoided. These are the reasons to transform the visual path-following problem in a reference tracking problem. The following expression represents the tracking error of the normalized tensor element $T_{221}$ with respect to a desired reference $T^d_{221}(t)$:

$$\zeta = T_{221} - T^d_{221}(t). \tag{4}$$

The normalization of the TT is done as defined at the end of Section 2 using $T_N = T_{232}$, which is non-null assuming that $\mathbf{C}_1 \neq \mathbf{C}_3$. The desired evolution of the tensor element is defined by the following differentiable sinusoidal reference:

$$
\begin{aligned}
T^d_{221}(t) &= \frac{T_{221}(0)}{2}\left(1+\cos\left(\frac{\pi}{\tau}t\right)\right),\ 0\leq t\leq\tau, \\
T^d_{221}(t) &= 0, \qquad\qquad\qquad\qquad t>\tau,
\end{aligned} \tag{5}
$$

where $T_{221}(0)$ is the initial value of the normalized tensor element or the value at the time of key image switching, and $\tau$ is a suitable time in which the tensor element must reach zero before the next switching of key image. Thus, the time is restarted at each instant when a change of key image occurs. Notice that because of the reference (5) has an initial value equals to the current value of $T_{221}$, the fed back error $\zeta$ at each key image switching is null, which avoids discontinuities. The proposed reference trajectory drives the tensor element to zero in a smooth way from its initial value.

The tracking error is computed using information extracted from the $i^{th}$ key image as $I_3(\mathbf{K},\mathbf{C}_3)$, the $(i-2)^{th}$ key image as $I_1(\mathbf{K},\mathbf{C}_1)$ and the current image $I_2(\mathbf{K},\mathbf{C}_2)$. According to the expressions of the trifocal tensor elements (3) and using the derivatives of the robot state as given by the model of the unicycle, the time-derivative of $T_{221}$ is obtained as follows:

$$
\begin{aligned}
\dot{T}^m_{221} &= -\dot{x}_2\cos\phi_2 + x_2\dot{\phi}_2\sin\phi_2 - \dot{y}_2\sin\phi_2 - y_2\dot{\phi}_2\cos\phi_2, \\
&= (x_2\sin\phi_2 - y_2\cos\phi_2)\,\omega = T^m_{223}\omega.
\end{aligned}
$$

This time-derivative is also valid for normalized tensor elements and therefore, the differential equation relating the rate of change of the error with the reference tracking (RT) velocity is as follows:

$$
\dot{\zeta} = T_{223}\omega_{rt} - \dot{T}^d_{221}. \tag{6}
$$

Thus, the velocity $\omega_{rt}$ is worked out from the error dynamics (6). The following rotational velocity assigns a new dynamics through the auxiliary input $\delta_a = -k_c\zeta$:

$$
\omega_{rt} = \frac{1}{T_{223}}\left(\dot{T}^d_{221} - k_c\zeta\right), \tag{7}
$$

where $k_c$ is a control gain. This velocity yields the error dynamics $\dot{\zeta} = -k_c\zeta$, which is exponentially stable for $k_c > 0$. This RT control is continuous with a sinusoidal behavior between key images. A nominal rotational velocity ($\bar{\omega}$) can be added in order to obtain an RT+ control that is maintained almost constant between key images, i.e., almost piece-wise constant rotational velocity during the navigation. So, the complete velocity can be eventually computed as:

$$
\omega = k_t\omega_{rt} + \bar{\omega}, \tag{8}
$$

where $k_t > 0$ is a weighting factor on the reference tracking control $\omega_{rt}$.

### *4.1 Nominal Velocity from the Memory*

Previous image-based approaches for navigation using a visual memory only exploit local information [4, 5, 6], i.e., the required rotational velocity is computed from the current and the nearest target images. It is proposed to exploit the visual memory to have an a priori information about the whole path without the need of a 3D reconstruction or representation of the path. A kind of qualitative map of the path can be obtained from the tensor element $T_{221}$ using three consecutive key images of the memory. The value of this element, denoted as $T^{ki}_{221}$, shows qualitatively the orientation of the camera in the $(i-1)^{th}$ key image with respect to the $i^{th}$ one and so, the nominal rotational velocity that appears in (8) can be computed from $T^{ki}_{221}$. Recall that the tensor is computed between all consecutive triplets of key images with target in the $i^{th}$ one. The nominal velocity $\bar{\omega}$ can be set proportional to the tensor elements $T^{ki}_{221}$ as follows:

$$
\bar{\omega} = \frac{k_m\upsilon}{d_{\min}}T^{ki}_{221}, \tag{9}
$$

where $k_m < 0$ is a constant factor. The factor $\upsilon/d_{\min}$ is needed to normalize the values of the tensor $T_{221}^{ki}$ for different conditions of a task regarding to the distance between key images. Although the translational velocity $\upsilon$ required in (9) can be set to a constant value, it is desirable to adapt this velocity according to the shape of the path, i.e., a large value of $T_{221}^{ki}$ implies that the curvature of the path is large, which demands to reduce the translational velocity. In any case, the value of the translational velocity is known. It is worth noting that the nominal rotational velocity by itself is able to steer the robot close to the path, but any deviation is corrected by (7) in (8).

### *4.2 Timing Strategy and Key Image Switching*

The proposed control method is based on taking to zero the tensor element $T_{221}$ before reaching the next key image, which imposes a constraint for the time $\tau$ of the reference. Thus, a strategy to define this time is related to the minimum distance between key images ($d_{\min}$) and the translational velocity ($\upsilon$) as follows:

$$\tau = \frac{d_{\min}}{\upsilon}.$$

A good compromise between this time and the settling time of the tracking error is to make $k_c = 12.5/\tau$. By using the controller (7) with the reference (5), the time $\tau$ and the control gain $k_c$ as described above, an intermediate location determined by $d_{\min}$ is reached. In the best case, when $d_{\min}$ coincides with the real distance between key images, the robot reaches the location where the corresponding key image was acquired. In order to achieve a good correction of the longitudinal position for each key image, the reference (5) is maintained to zero, which implies that $\omega = 0$, until the image error starts to increase. The image error is defined as the mean squared error between corresponding image points of the current image ($\mathbf{p}_{i,j}$) and points of the next closest target key image ($\mathbf{p}_j$), i.e.:

$$\epsilon = \frac{1}{r}\sum_{j=1}^{r} \left\| \mathbf{p}_j - \mathbf{p}_{i,j} \right\|, \tag{10}$$

where $r$ is the number of used corresponding points. As shown in [16], the image error decrease monotonically until the robot reaches each target view. In this work, the increment of the image error is used as the switching condition for the next key image, which is confirmed by using the current and previous difference of instantaneous values of the image error.

## 5. FUZZY CONTROL FOR VISUAL PATH-FOLLOWING

Taking advantage of the experience in the memory-based visual control of a mobile robot from the TT obtained from the results reported in [17], Mamdani-type fuzzy rules are designed in this section to drive a mobile robot following a visual path. This type of controller has an heuristic nature [21] that can be exploited in a visual path-following task in order to mimic the human action as in other applications, e.g. helicopters control [22]. First, as described at the end of Section 4.1, it is desirable to set the translational velocity according to the curvature of the path, which can be carried out by a fuzzy inference system using $T_{221}^{ki}$ as input. Second, and more important, to reach the final location, the robot must be steered appropriately to move from one key image to the next, essentially by aligning its heading to the next key image. Given that the TT provides a direct measurement of the disalignment with respect to the path through the current value of $T_{221}$, a set of heuristic rules can be designed to achieve the correction. The nature of the problem is to provide an adequate value of the rotational velocity depending on the visual measurement. There is no need of modeling a complex system as in the Takagi-Sugeno fuzzy approach [11], or dealing with uncertainties in an estimation process as in fuzzy-type stochastic approaches [13]. Therefore, Mamdani-type fuzzy control is selected as an adequate control technique for the problem. This type of fuzzy control has been previously used for visual lane-following in [10], which is a similar problem to the one addressed in this paper. However, the use of a visual memory instead of an artificial lane is more natural and flexible in the sense that no modification of the environment is needed.

## 5.1 *Adapting the Translational Velocity*

The a priori information available in the sequence of key images can be exploited to adapt the translational velocity according to the shape of the path encoded in $T_{221}^{ki}$. The values of $T_{221}^{ki}$ are obtained for each triplet of consecutive key images from the set of images that defines the visual path. For each segment between key images, these values give the notion of the path's curvature. Therefore, a decision about an adequate translational velocity can be derived from $T_{221}^{ki}$. A single-input-single-output fuzzy inference system is proposed to adapt the translational velocity.

- *Fuzzy representation:* Fig. 4(a) shows the input membership functions for the fuzzy inference system that receive the magnitude of the values $T_{221}^{ki}$ normalized by the minimum distance $d_{\min}$, i.e., $|\bar{T}_{221}^{ki}|$, where $\bar{T}_{221}^{ki} = T_{221}^{ki}/d_{\min}$. The normalization is needed to have similar values of $T_{221}^{ki}$ for different conditions of distance between key images. Three membership functions are used, corresponding to negative, zero and positive values of the input.

  The output membership functions of the inference system are presented in Fig. 4(b). As shown below in the fuzzy rules, one output membership function is used for each rule. The range of the output membership functions is a user-defined parameter. In the figure, the minimum translational velocity is defined to be 0.2 m/s and the maximum 0.4 m/s.

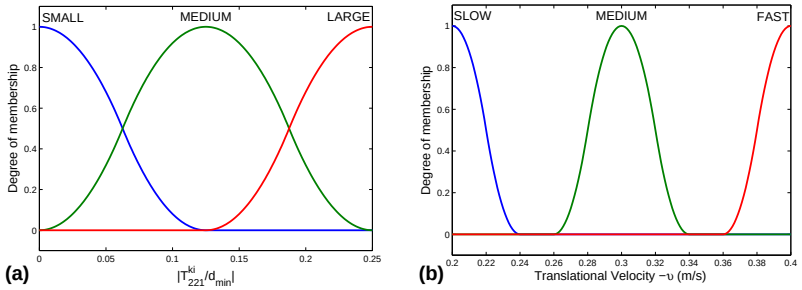


**Figure 4: Membership functions for inference of the translational velocity. (a) Input $|\bar{T}_{221}^{ki}|$. (b) Output $\upsilon$.**

- *Fuzzy rules:* Given the magnitude of the normalized tensor between key images, the fuzzy system infers an adequate translational velocity by using the following rules:

  1. If $|\bar{T}_{221}^{ki}|$ is SMALL, then $\upsilon$ is FAST.
  2. If $|\bar{T}_{221}^{ki}|$ is MEDIUM, then $\upsilon$ is MEDIUM.
  3. If $|\bar{T}_{221}^{ki}|$ is LARGE, then $\upsilon$ is SLOW.

For instance, the interpretation of the first rule is that if the value of $|\bar{T}_{221}^{ki}|$ is small (meaning that the curvature of the path is small in the segment), then, the translational velocity can be large and the robot moves fast. Notice that for each visual path, there are as much values of $|\bar{T}_{221}^{ki}|$ as key images, and the output given by the fuzzy system is changed at each key image switching. In order to smooth these transitions a sigmoidal function can be used.

## 5.2 *Rotational Velocity Controller for Reference Tracking*

Under the same framework described in Section 4, where the visual path-following problem is treated as a reference tracking problem, in this section a fuzzy controller is proposed to accomplish the navigation task. As introduced previously, the problem of following a sequence of key images can be faced by moving forward and rotating appropriately when an error expressed in terms of image data increases. The control goal for each segment between key images is to drive $T_{221}$ to zero before reaching the next key image. To achieve that, the same timing strategy and the condition of key image switching detailed in Section 4.2 is used for the fuzzy control. The proposed two-input-one-output fuzzy controller is able to solve the reference tracking problem described in Section 4. Thus, an RT control is obtained from the fuzzy inference system described next.

- *Fuzzy representation:* Given that a reference tracking problem is faced, the inputs of the fuzzy controller are the tracking error $\zeta$ (4) and its time-derivative $\dot{\zeta}$ (6). Fig. 5(a) displays the proposed input membership functions for $\zeta$. To avoid the input to leave the membership functions range, trapezoidal functions with large range are used. In Fig. 5(b) the input membership functions for $\dot{\zeta}$ are shown. As the robot motion is smooth with a reference tracking control, the range of $\dot{\zeta}$ is small. Membership functions corresponding to negative, zero and positive values are proposed for both inputs.

  Fig. 5(c) presents the proposed output membership functions. One output membership function is used for each rule. The labels for each output membership function are from W1 to W9, representing the possible values for the RT control. For example, W1 represents the largest negative value, W9 the largest positive value and W5 is the membership function for zero output.

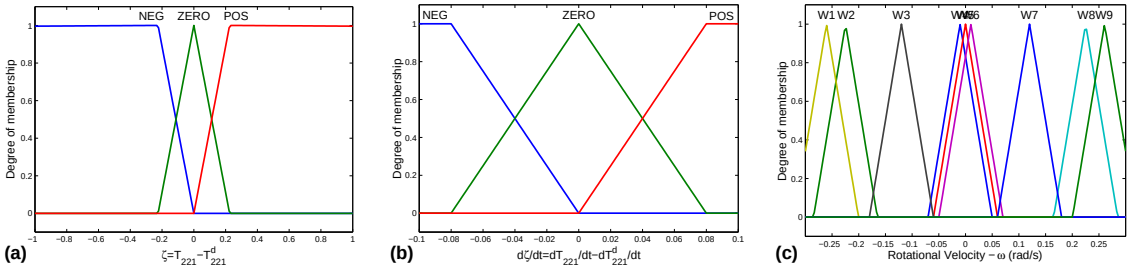


**Figure 5: Membership functions for the reference tracking controller. (a) Input $\zeta$. (b) Input $\dot{\zeta}$. (c) Output $\omega$.**

- *Fuzzy rules:* Given a value of the tracking error and its time-derivative, the fuzzy inference system provides a rotational velocity to keep the error around zero by using the following nine rules:

  1. If $\zeta$ is NEG and $\dot{\zeta}$ is NEG, then $\omega_{rt}$ is W1.
  2. If $\zeta$ is ZERO and $\dot{\zeta}$ is NEG, then $\omega_{rt}$ is W2.
  3. If $\zeta$ is NEG and $\dot{\zeta}$ is POS, then $\omega_{rt}$ is W3.
  4. If $\zeta$ is NEG and $\dot{\zeta}$ is ZERO, then $\omega_{rt}$ is W4.
  5. If $\zeta$ is ZERO and $\dot{\zeta}$ is ZERO, then $\omega_{rt}$ is W5.
  6. If $\zeta$ is POS and $\dot{\zeta}$ is ZERO, then $\omega_{rt}$ is W6.
  7. If $\zeta$ is POS and $\dot{\zeta}$ is NEG, then $\omega_{rt}$ is W7.
  8. If $\zeta$ is ZERO and $\dot{\zeta}$ is POS, then $\omega_{rt}$ is W8.
  9. If $\zeta$ is POS and $\dot{\zeta}$ is POS, then $\omega_{rt}$ is W9.

These rules are derived heuristically by analyzing the behavior of the RT control of equation (7) reported in [17]. The interpretation of the first rule is as follows: if the tracking error is negative and its negativeness is increasing because the time-derivative of the error is negative, then, a large negative rotational velocity is required to reduce the error. The other rules can be similarly explained. Notice that the RT control given by this fuzzy controller can be used alone to accomplish the visual path-following task or can be combined with (9) as defined in (8) to obtain a piece-wise rotational velocity.

### *5.3 Integrating Nominal Velocity and Deviation Correction in a Single Fuzzy Controller*

In this section, a novel fuzzy controller for visual path-following is presented. Based on the idea of the RT+ control given by (8), this new approach integrates the two components of this control in a single fuzzy controller. This means that instead of adding the nominal velocity and the component that corrects deviation as in (8), both components are integrated by a fuzzy inference system to generate the required rotational velocity for path-following. The controller relies on the timing strategy and switching condition of key images described in Section 4.2.

- *Fuzzy representation:* This controller takes the normalized tensor element between key images $\bar{T}_{221}^{ki}$ and the current value of the tensor element $T_{221}$ as inputs. Fig. 6(a) and Fig. 6(b) show the membership functions for the Mamdani controller for both inputs. The membership functions are similar for each input and their range is defined large enough. The proposed membership functions correspond to

negative, zero and positive values for both inputs. Fig. 6(c) presents the output membership functions of the controller. One membership function is used for each rule and the values are defined on the basis of the results with the nonlinear control of Section 4 [17].

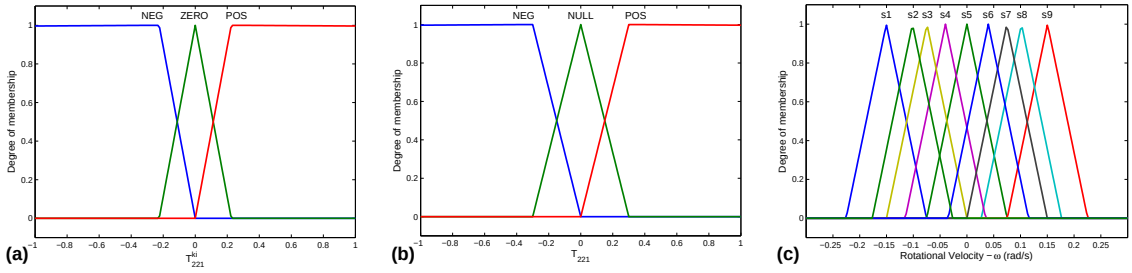


**Figure 6: Membership functions for the single fuzzy controller. (a) Input $\bar{T}_{221}^{ki}$. (b) Input $T_{221}$. (c) Output $\omega$.**

- *Fuzzy rules:* Nine rules are proposed for this controller and one output membership function is used for each rule. The fuzzy inference system gives an adequate rotational velocity to correct any deviation from the visual path by taking $\bar{T}_{221}^{ki}$ and $T_{221}$ as described in the following cases:

1. If $\bar{T}_{221}^{ki}$ is NEG and $T_{221}$ is NEG, then $\omega$ is s4.
2. If $\bar{T}_{221}^{ki}$ is ZERO and $T_{221}$ is NEG, then $\omega$ is s3.
3. If $\bar{T}_{221}^{ki}$ is NEG and $T_{221}$ is POS, then $\omega$ is s8.
4. If $\bar{T}_{221}^{ki}$ is NEG and $T_{221}$ is ZERO, then $\omega$ is s9.
5. If $\bar{T}_{221}^{ki}$ is ZERO and $T_{221}$ is ZERO, then $\omega$ is s5.
6. If $\bar{T}_{221}^{ki}$ is POS and $T_{221}$ is ZERO, then $\omega$ is s1.
7. If $\bar{T}_{221}^{ki}$ is POS and $T_{221}$ is NEG, then $\omega$ is s2.
8. If $\bar{T}_{221}^{ki}$ is ZERO and $T_{221}$ is POS, then $\omega$ is s7.
9. If $\bar{T}_{221}^{ki}$ is POS and $T_{221}$ is POS, then $\omega$ is s6.

The interpretation of the first rule is that if $\bar{T}_{221}^{ki}$ is negative, meaning that the robot must rotate in positive sense according to the key images, and if $T_{221}$ is also negative, meaning that the robot is currently rotating in the desired positive sense, then the rotational velocity is small negative to keep a small rotation in the current sense, which is the correct action. All the other rules can be similarly explained.

Notice that this controller provides the complete required velocity to keep the robot following the path or to steer it into the path if it is not. Besides, as the inputs are direct measurements obtained from the images, this controller presents good robustness against image noise.

## 6. EVALUATION OF THE FUZZY VISUAL CONTROL SCHEME

In this section, Matlab simulations of the proposed navigation scheme are presented. The generic camera model [18] is used to generate synthetic key images from the 3D scene of Fig. 7(a) according to the robot motion on the predefined path shown in the same figure. This learned path starts in the location (5,-5,0$^o$) and finishes just before to close the loop of 54m long. The vision system is hypercatadioptric with parameters $\alpha_x = 950$, $\alpha_y = 954$, $x_0 = 512$, $y_0 = 384$ all of them in pixels, $\xi = 0.9662$ and the size of the images is 1024×768 pixels. The TT is estimated using the typical 7-point algorithm introduced in Section 2.2 and using the projected points on the sphere. Fig. 7(b) shows an example of a triplet of catadioptric images projected onto the unitary sphere. Regarding to the fuzzy inference systems, they are implemented using the Fuzzy Logic Toolbox of Matlab with the following features: product for AND method, maximum for OR method, product for implication, sum for aggregation and centroid for defuzzification.

First, the performance of the proposed fuzzy controller for reference tracking detailed in Section 5.2 is evaluated. The robot task is to follow a visual path of 36 key images distributed randomly along the learned path. The random distance between consecutive key images is between 1.42m and 1.6m, in such a way that a minimum distance $d_{\min} = 1.4$m is assumed. The translational velocity is bounded between $0.2$m/s and $0.4$m/s. Results for two cases are presented: 1) only reference tracking (RT control) given by the fuzzy controller of Section 5.2, and 2) reference tracking + nominal velocity (RT+) as given by (8).

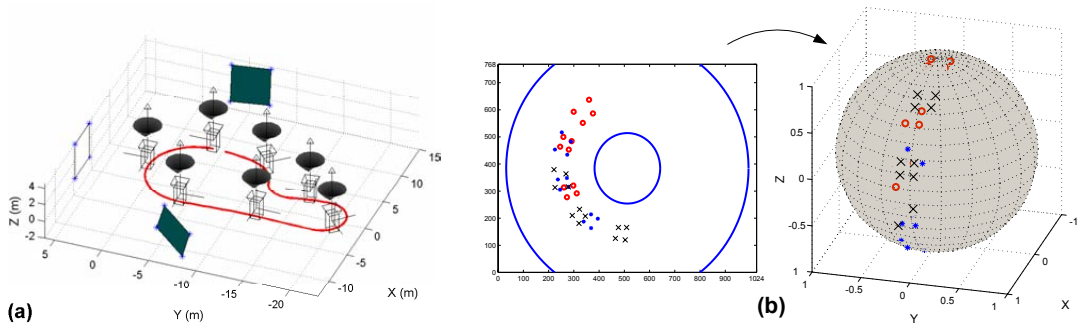


**Figure 7: Virtual scene and example of the synthetic images used. (a) Tridimensional scene and predefined path. (b) Example of a triplet of images projected onto the unitary sphere.**

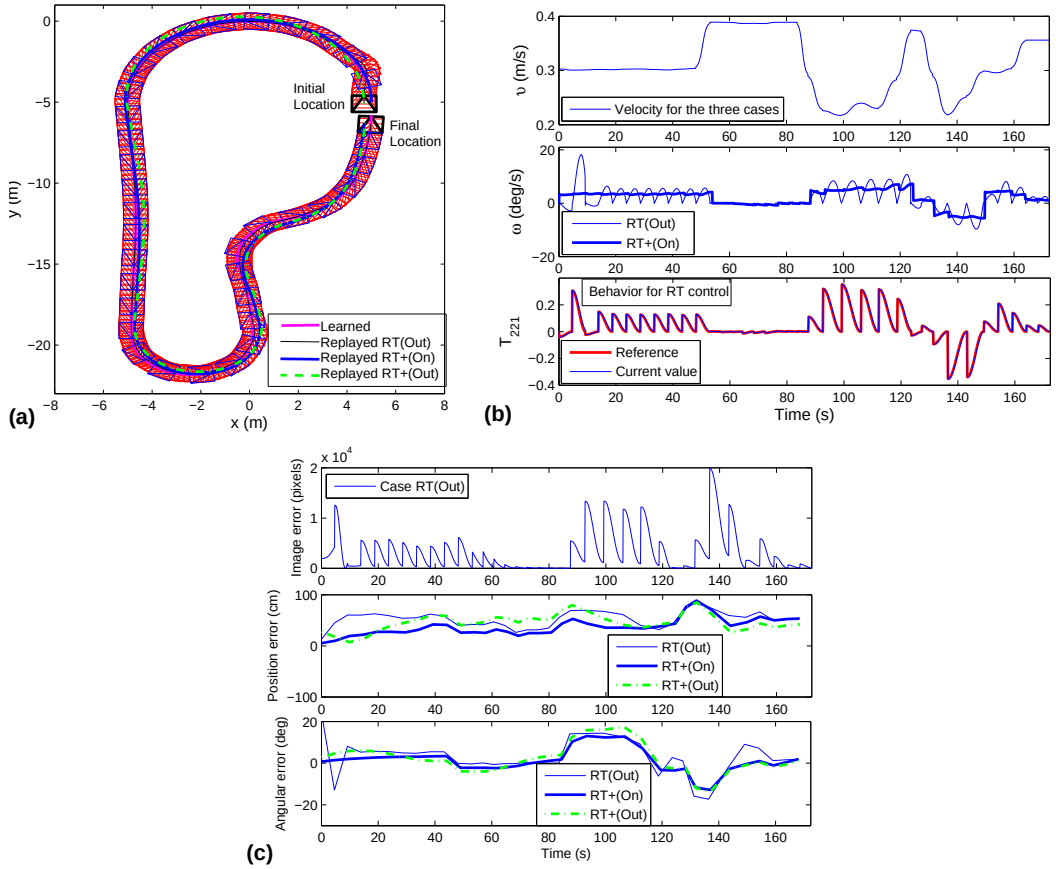


**Figure 8: Simulation results for a navigation task using the fuzzy controller for reference tracking and nominal velocity for 36 key images. (a) Resultant paths and key images distribution. (b) Velocities and evolution of the element $T_{221}$. (c) Image error and path-following errors.**

It can be seen in Fig. 8(a) that the resultant path of the autonomous navigation stage is almost similar to the learned one in both cases; however, as expected, the performance is better for the RT control that is able to steer the robot into the path if it starts out of path. The RT+ control is only able to drive the robot close to the path in such condition. The first plot of Fig. 8(b) shows how the translational velocity effectively

changes according to the shape of the path. For instance, between 55 s and 85 s the higher velocity is applied, which corresponds to the almost straight part of the path.

The second plot of Fig. 8(b) shows the behavior of the rotational velocity. On one hand, the velocity given by the RT control is smooth, performing cycles that start from zero and return to zero for each key image. On the other hand, the piece-wise constant velocity given by the RT+ control is more natural. The rotational velocity given by the RT+ when the robot starts out of the path is almost the same to the one given by the RT+ starting on the path and it is not presented. The third plot of Fig. 8(b) shows the behavior for the reference tracking of $T_{221}$. It can be seen that the tensor element does not present unstable behavior when a key image is reached ($T_{221}$ reaches zero), which means that the problem of short baseline is not present. Additionally, the current value of $T_{221}$ tracks the desired reference (5) really close, which is enough to solve effectively the visual path-following problem.

The performance of the approach for the same experiment is presented in Fig. 8(c). The first plot of the figure shows the behavior of the image error for the RT case. It can be seen that the image error exhibits a monotonic decay before reaching each key image. Only the image error for the RT control with the robot starting out of the path is shown given that a similar behavior can be seen for each case. The largest peaks in the image error correspond to the sharp curves in the path, which also causes the highest error in the path-following, as can be seen in the second and third plot of Fig. 8(c). Nevertheless, the path-following errors to reach each key image are small and comparable for both controllers.

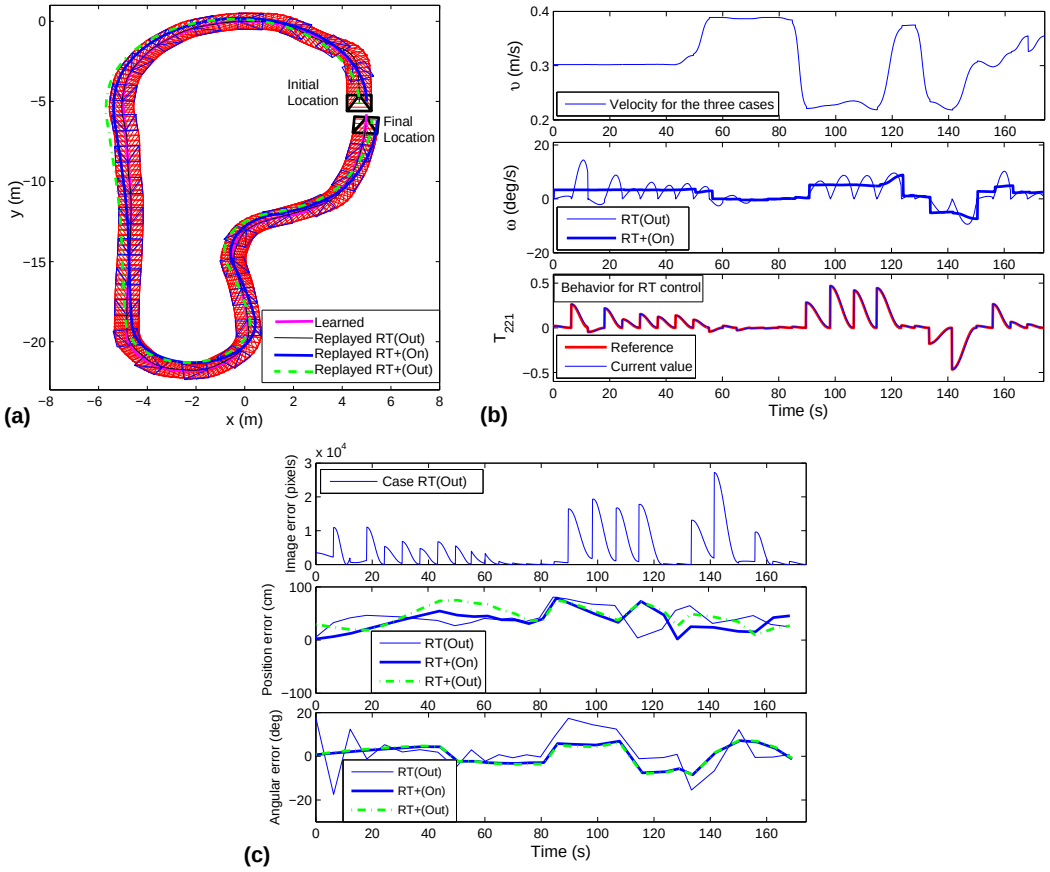


**Figure 9: Simulation results for a navigation task using the fuzzy controller for reference tracking and nominal velocity for 28 key images. (a) Resultant paths and key images distribution. (b) Velocities and evolution of the element $T_{221}$. (c) Image error and path-following errors.**

In order to evaluate the performance of the scheme using harder conditions, 28 key images are placed randomly along the predefined path separated from 1.8m to 2.0m. Therefore, a minimum distance $d_{\min} = 1.75$m is assumed. The path-following is really good along the whole path for the RT control and reasonably good for the RT+ (Fig. 9(a)). The RT+ control is slightly sensitive to longer distance between key images along sharp curves. Although the number of key images is less than in the previous simulation, Fig. 9(b) presents a similar behavior of the robot velocities with respect to the results for 36 key images. As can be seen in Fig. 9(c), the image error is in general larger than in Fig. 8(c) due to the larger distance between key images. However, the errors to reach each key image are still comparable for the three cases shown.

Next, the fuzzy controller described in Section 5.3 is evaluated and compared with the nonlinear controller introduced in Section 4 and detailed in [17]. In this case the simulated vision system is changed for a paracatadioptric one. The size of the images is 800x600 pixels. Additionally, Gaussian noise of standard deviation of 1 pixel is added to the synthetic images. In Fig. 10(a) can be seen that the navigation is carried out with good performance by using the single fuzzy controller even for the presented case where the robot starts out of the path. Although the nonlinear controller also achieves to accomplish the navigation task, the path is better followed using the fuzzy controller. The effect of the image noise can be seen in the robot velocities shown in Fig. 10(b), however, it does not affect significantly the rotational velocity given by the fuzzy controller as in the case of epipolar control [9] or nonlinear control with the TT [17], where large peak values of the velocity may appear. As appreciated in Fig. 10(c), the estimation of the current value of $T_{221}$ is not free of noise for both controllers, however, the fuzzy controller is less affected. The filtering property of the TT as a geometric constraint is useful to mitigate the effect.

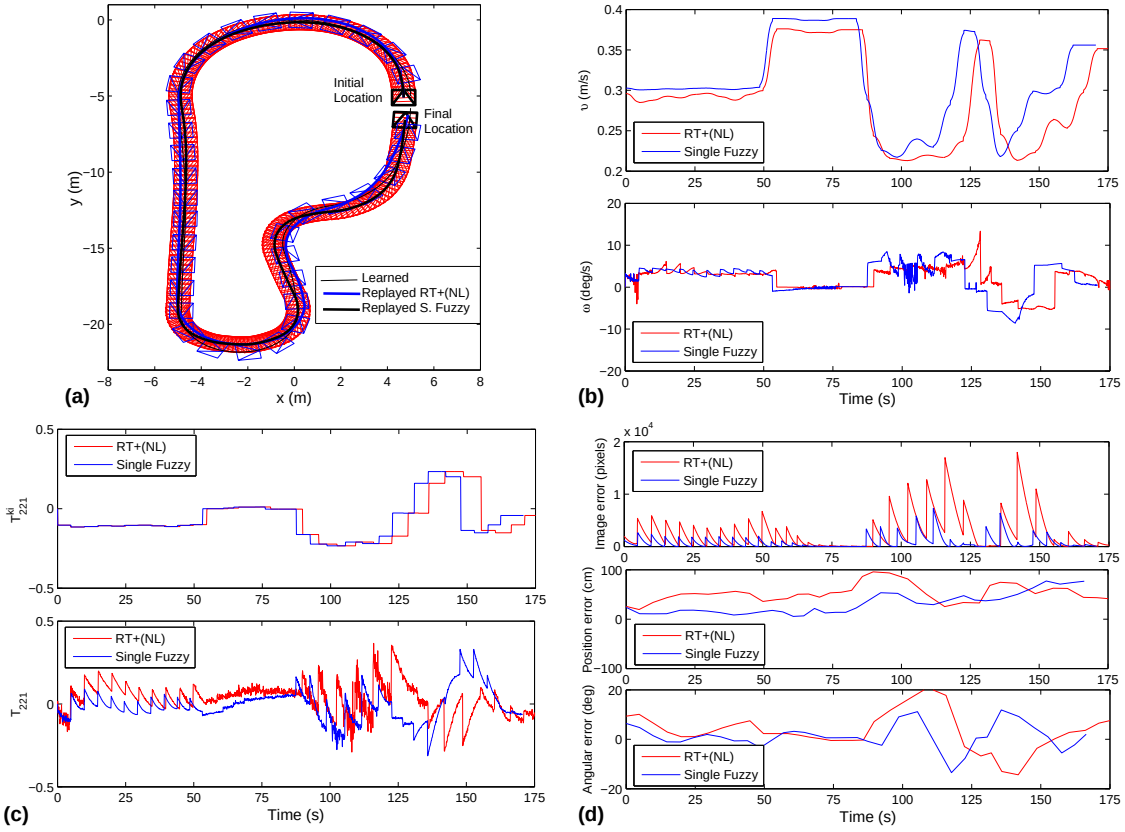


**Figure 10: Simulation results for a navigation task using the single fuzzy controller in comparison to the nonlinear controller in [17]. (a) Resultant paths and key images distribution. (b) Robot velocities. (c) Controllers inputs. (d) Image errors and path-following errors.**

A quantitative comparison between the fuzzy and nonlinear controllers is shown in Fig. 10(d). A lower image error can be seen for the fuzzy controller during all the navigation. Also, the position and angular errors to reach each key image show the good performance of the fuzzy controller in the path-following task. It is worth emphasizing that the design of the single fuzzy controller is also a good contribution from a theoretical point of view, because such controller does not require the use of a time-varying reference to be tracked in contrast to the nonlinear controller presented in (8) and evaluated previously in [17]. Thus, the fuzzy controller is simple and effective for the goal of computing a smooth or piece-wise constant rotational velocity.

In order to show the behavior of the visual information, Fig. 11 presents two examples of the motion of the image points along the whole navigation. Although 12 points are used to compute the tensor, only the motion of 7 points is shown. It is appreciable the advantage of using a central catadioptric system looking upward, which is able to see the same scene during the whole navigation task.

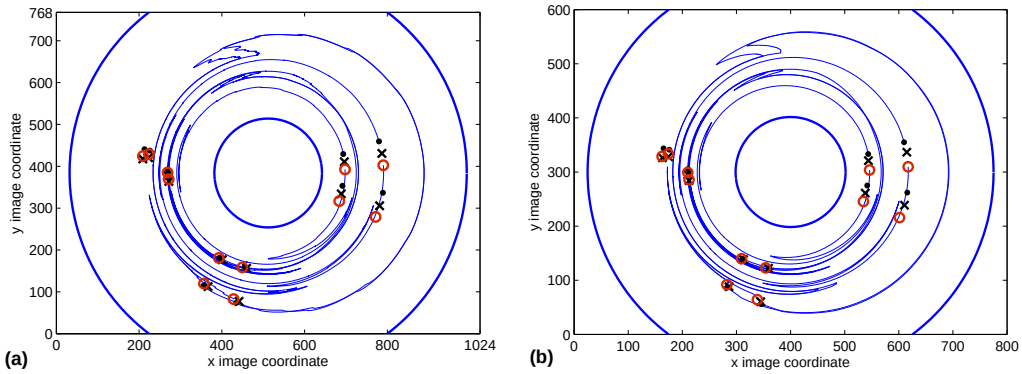


**Figure 11: Motion of the points in the images along the navigation for (a) case of Fig. 8 and (b) case of Fig. 10. Markers: "·" initial image, "O" final key image, "×" image at final reached location.**

## 7. CONCLUSIONS

In this paper, a fuzzy control scheme for wheeled mobile robot navigation based on a visual memory has been proposed. The value of one element of the trifocal tensor computed from image points is the unique required information by the approach. The proposed image-based scheme does not need pose parameters decomposition. In this context, the scheme avoids discontinuous rotational velocity when a new target image must be reached providing piece-wise constant velocities as desired. The translational velocity is adapted according to the curvature of the path by a fuzzy inference system and the approach is independent of this velocity. To the author's knowledge, it is the first time that fuzzy logic is exploited for visual memory-based navigation. The use of fuzzy control has achieved an effective and simpler controller than the nonlinear controller in [17] by avoiding the need of a time-varying reference to be tracked. Additionally, the proposed scheme is valid for any central camera (from conventional to wide field of view cameras), which increases its applicability. The control scheme has presented good performance according to simulation results using synthetic images.

## REFERENCES


[1] F. Chaumette and S. Hutchinson. Visual servo control part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(14):82–90, 2006.

[2] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[3] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 317–322, 1999.
[4] T. Goedeme, T. Tuytelaars, L. V. Gool, G. Vanacker, and M. Nuttin. Feature based omnidirectional sparse visual path following. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1806–1811, 2005.
[5] E. Royer, M. Lhuillier, M. Dhome, and J. M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.
[6] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Trans. on Intel. Transportation Systems*, 10(3):392–402, 2009.
[7] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette. Outdoor visual path following experiments. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4265–4270, 2007.
[8] A. Cherubuni and F. Chaumette. Visual navigation with a time-independent varying reference. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 5968–5973, 2009.
[9] H. M. Becerra, J. Courbon, Y. Mezouar, and C. Sagüés. Wheeled mobile robots navigation from a visual memory using wide field of view cameras. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5693–5699, 2010.
[10] G. Antonelli, S. Chiaverini, and G. Fusco. A fuzzy-logic-based approach for mobile robot path tracking. *IEEE Transactions on Fuzzy Systems*, 15(2):211–221, 2007.
[11] C-S. Tseng, B-S. Chen, and H-J. Uang. Fuzzy tracking control design for nonlinear dynamic systems via ts fuzzy model. *IEEE Transactions on Fuzzy Systems*, 9(3):381–392, 2001.
[12] A-M. Zou, Z-G. Hou, Z-Q. Cao, and M. Tan. Robust passivity-based adaptive control of a nonholonomic mobile robot using fuzzy logic. *Intel. Automation and Soft Computing*, 15(2):187–200, 2009.
[13] C. S. Chan and H. Liu. Fuzzy qualitative human motion analysis. *IEEE Transactions on Fuzzy Systems*, 17(4):851–862, 2009.
[14] I. H. Suh and T. W. Kim. Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators. *IEEE Transactions on Fuzzy Systems*, 2(3):203–220, 1994.
[15] T. Kato. Trajectory tracking of four-wheel steered mobile robots by an image-based fuzzy controller. In *IEEE International Symposium on Industrial Electronics*, pages 1546–1550, 2012.
[16] Z. Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 2686–2692, 2006.
[17] H. M. Becerra and C. Sagüés. Visual control for memory-based navigation using the trifocal tensor. In *World Automation Congress*, pages 1–6, 2012.
[18] C. Geyer and K. Daniilidis. An unifying theory for central panoramic systems and practical implications. In *European Conf. on Computer Vision*, pages 445–461, 2000.
[19] G. López-Nicolás, J.J. Guerrero, and C. Sagüés. Visual control through the trifocal tensor for nonholonomic robots. *Robotics and Autonomous Systems*, 58(2):216–226, 2010.
[20] H. M. Becerra, G. López-Nicolás, and C. Sagüés. Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. *Robotics and Autonomous Systems*, 58(6):796–808, 2010.
[21] K. Passino and S. Yurkovich. *Fuzzy Control*. Addison-Wesley-Longman, M. P., CA, USA, 1998.
[22] E. Sánchez, H. M. Becerra, and C. M. Vélez. Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. *Information Sciences*, 177(10):1999–2022, 2007.