Chapter 8

Predicting Collisions in Mobile Robot Navigation by Kalman Filter

Angel Sánchez, Homero Ríos,
Gustavo Quintana and Antonio Marín

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.71653

Abstract

The growing trend of the use of robots in many areas of daily life makes it necessary to search for approaches to improve efficiency in tasks performed by robots. For that reason, we show, in this chapter, the application of the Kalman filter applied to the navigation of mobile robots, specifically the Time-to-contact (TTC) problem. We present a summary of approaches that have been taken to address the TTC problem. We use a monocular vision-based approach to detect potential obstacles and follow them over time through their apparent size change. Our approach collects information about obstacle data and models the behavior while the robot is approaching the obstacle, in order to predict collisions. We highlight some characteristics of the Kalman filter applied to our problem. Finally, we show of our results applied to sequences composed of 210 frames in different real scenarios. The results show a fast convergence of the model to the data and good fit even with noisy measures.

Keywords: Kalman filter, Time-to-contact, avoiding collisions, robot navigation, forecasting

1. Introduction

Nowadays, many robotic entities have been developed in order to make many human activities more efficient. In fact, it is common to find robots in hospitals, factories, and even in homes, which help to automate many tasks autonomously or semi-autonomously. However, many problems can arise when mobile autonomous robots must travel in uncertain and unknown environments. For this reason, techniques, programs, and sensors have been developed over time to address such tasks as localization, communication, path planning, and collision avoidance. Faced with these problems, robots must be able to sense their environment and understand it to make the best decisions. From these tasks, we will analyze the task of avoiding obstacles by mobile robots, using the well-known term "Time-to-contact" or TTC.

TTC is a biologically inspired method for obstacle detection and reactive control of motion. TTC was first studied and defined in [1] as "*the distance to an obstacle divided by the relative velocity between them*." In other words, TTC is the elapsed time before an observer (the center of projection) makes contact with the surface being viewed if the current relative motion between the observer (e.g., a robot's camera) and the surface were to continue without changes, i.e., under constant relative velocity. TTC is usually expressed in terms of the speed and the distance of the considered obstacle. The classical equation to compute TTC is given by Eq. (1)

$$TTC = -\frac{Z}{\frac{dZ}{dt}} \tag{1}$$

where Z is the distance between the observer and the obstacle, and $\frac{dZ}{dt}$ is the velocity of the robot with respect to the obstacle. **Figure 1** shows the camera model and perception of obstacles from a mobile robot using monocular vision, where *t* represents the time, *Z* is the distance, *f* referring to the focal length, *r* is defined as the distance between the center of projection and the obstacle, and S representing the height of an obstacle.

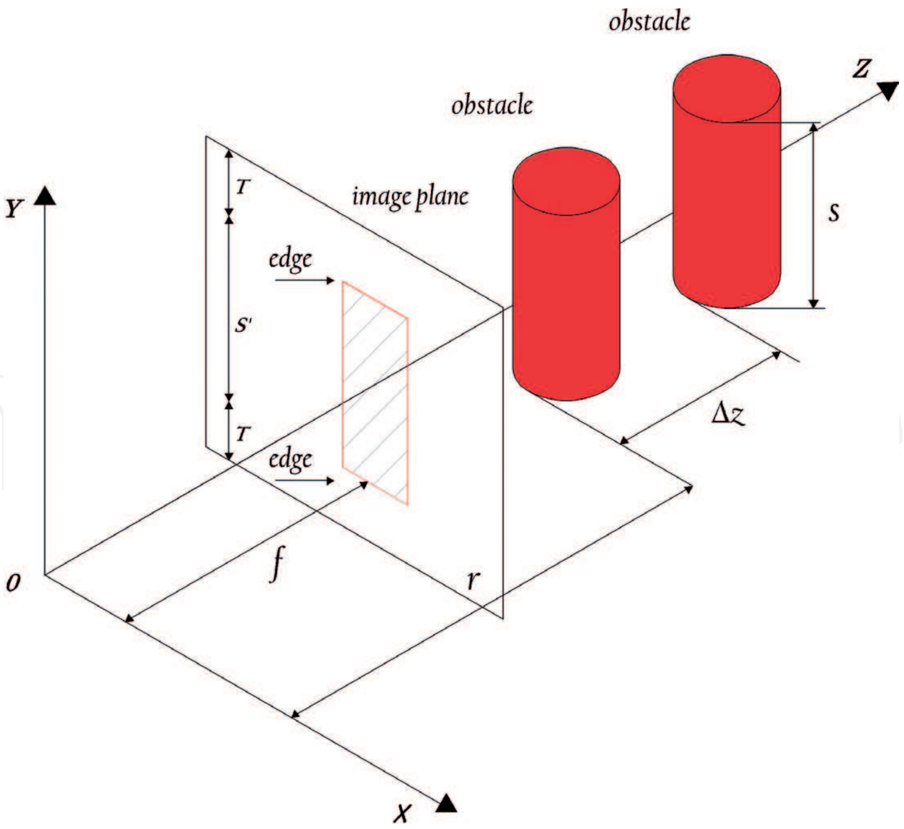

Figure 1. Isometric view of the model of perception.

The main idea of this chapter is to address this problem, modeling the growth of objects as the camera approaches the obstacle, in order to avoid detection obstacles in each frame once a precise model is built. First, we highlight some important works in the literature to give an overview of the different approaches that have been used to address this problem, discussing advantages and disadvantages of them. In the next section, we will give a brief description of the whole process that we use and how we address the problem using the Kalman Filter. Subsequently, we describe some experiments and results that we have made, to conclude with some conclusions and future work.

2. Background

The problem of estimating the Time-to-contact continues to be approached with different techniques. This is due to the fact that there are some factors that can prevent the robot from recognizing its environment reliably, such as the change of light intensity in the environment, erroneous segmentation of obstacles, or errors due to sinuous floors (especially with mobile robots on wheel).

Perhaps, the most practical and accurate approach is to use specialized sensors embedded in robots, such as sonars [2, 3]. Depth sensing used in the literature for robotic navigation and egomotion has been performed by binocular vision (stereo vision) [4, 5]. Other studies [6] have estimated TTC using paracatadioptric sensors with good results in real time. Although ultrasonic range sensors have large field of view, cross-talk problems might appear if more than one sensor is used simultaneously. As a result, "*the frequency of obstacle detection is limited by the number of sensors in use and the time required for an echo to return from an obstacle*" [7].

In order to minimize the energy consumption and cost, we decided to use passive sensors such as one camera only, i.e., monocular vision. Several studies have employed monocular vision to estimate TTC. For example, motion has been computed from images in space and spectral domains, and specifically TTC has been estimated using temporal change in power spectra between successive images [8]; however, switching from time domain to frequency domain involves extra processing. Additionally, TTC can be estimated for different goals (e.g., docking and landing) [9, 10] from the focus of expansion (FOE), however, these approaches are based on estimating optical flow, and hence they "*are iterative, need to work at multiple scales, tend to be computationally expensive and require a significant effort to implement properly*" [11].

Other method to estimate TTC is by "Direct method," which works directly with the derivatives of image brightness and does not require feature detection, feature tracking, or estimate of the optical flow [12]. Despite this method has achieved good results to approach surfaces, there are cases where the accuracy is compromised (e.g., when the robot approaches untextured walls and thus changing of the brightness is zero).

Finally, TTC has been computed using changes in the obstacle's size. For instance, studies in [13, 14] have used the fact that animals and insects obtain information from the apparent size S of objects and the temporal changes in the size. This information is usually called the

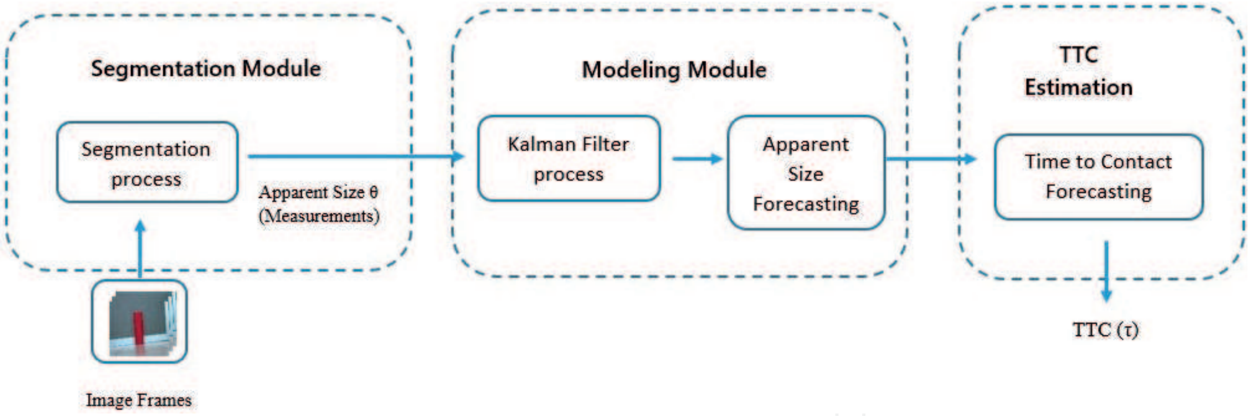

Figure 2. Methodology to performance the TTC forecasting.

"tau-margin" defined as Eq. (2). Tau-margin is derived from Eq. (1) by using a characteristic size of the obstacle on the image [15] and the approximation that the obstacle is planar and parallel to the image plane.

$$\tau = -\frac{S}{\frac{dS}{dt}} \tag{2}$$

However, a last focus on which we are interested is based on modeling the robot's movement. The study [16] has calculated TTC in vehicular motion using several scenarios. The idea is promising; however, due to the use of interest points, a mechanism needs to be implemented for grouping these points into different regions representing different obstacles.

In this brief summary, we can see that many approaches have been proposed, however, more approaches are still emerging because there are factors that condition the accuracy of the TTC estimate. But, we decided to use monocular vision to minimize costs of specialized sensors (in terms of energy and money).

The approach we have worked on is to model the apparent sizes of the segmented obstacle in each number of frames, in order to predict the Time-to-contact, that is, in how many frames the robot could collide with that obstacle. Since the TTC is estimated by analyzing how the apparent size of the object is changing with respect to time, we can find errors if the segmentation is not correct, which would lead to an incorrect acceleration or deceleration of the robot and to a possible collision.

When constructing models of a phenomenon, we base the predictions on previous data, which leads the robot to "understand" the behavior of its environment, if it has a constant speed. Having explained the importance of modeling, we decided to incorporate the Kalman Filter to address this problem and have reliable predictions. **Figure 2** shows the methodology used to estimate and forecast TTC.

3. Estimating Time-to-contact

In the following sections, we will give an outline of the processes used in each module, shown in **Figure 2**.

3.1. Segmentation process

In a controlled experimental environment, we detect obstacles by color, and analyze the height of the segmented region by enclosing it in a rectangle. The colors are pre-calibrated. **Figure 3** shows examples of two experimental cases and their segmentation by color. In the first scenario, the robot approaches an opaque red cylinder. In the second case, the robot approaches a bright green sphere.

3.2. Calculating apparent size

Figure 4 represents a mobile robot model, where S is representing the size of the detected object, r represents the distance between the mobile robot and the detected obstacle, and θ the proportional the aperture of the angle to the object's size projected on the image called "apparent size."

Figure 5 illustrates how the apparent size of the objects is expanded while it is approaching the camera. From a perspective view, T regions decrease in the image and the apparent size S′ (S projected on the image) grows proportionally as the robot approaches the obstacle. The

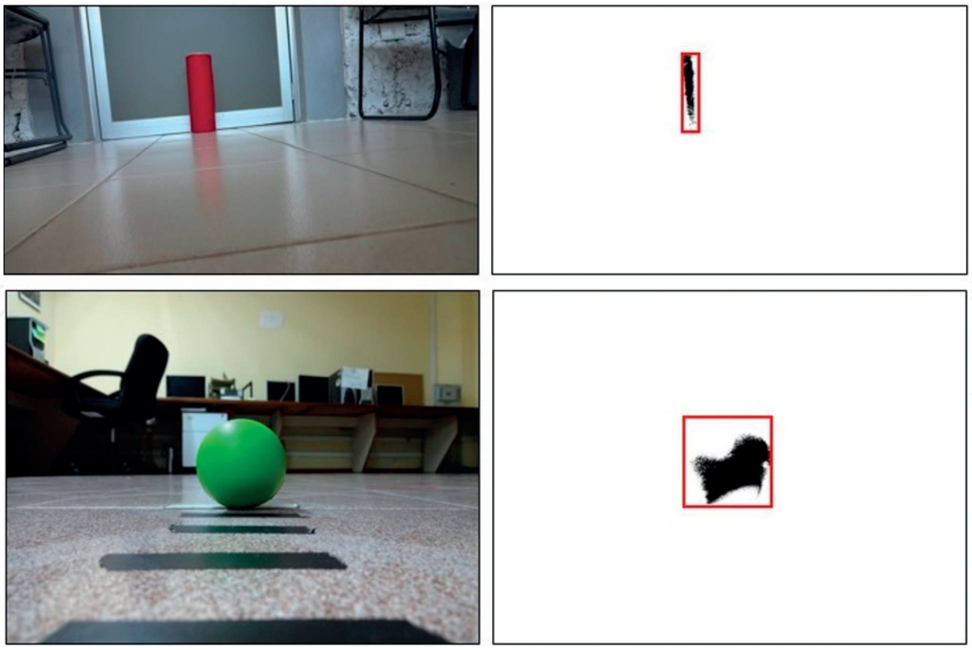

Figure 3. Example of segmentation by color process with experimental scenarios.

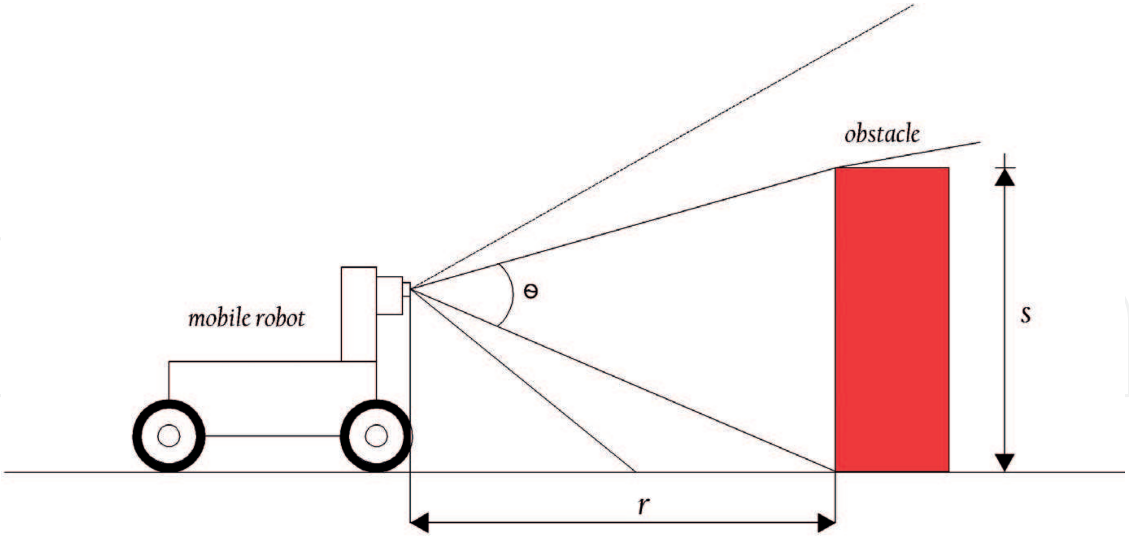

Figure 4. View of the mobile robot approaching an obstacle.

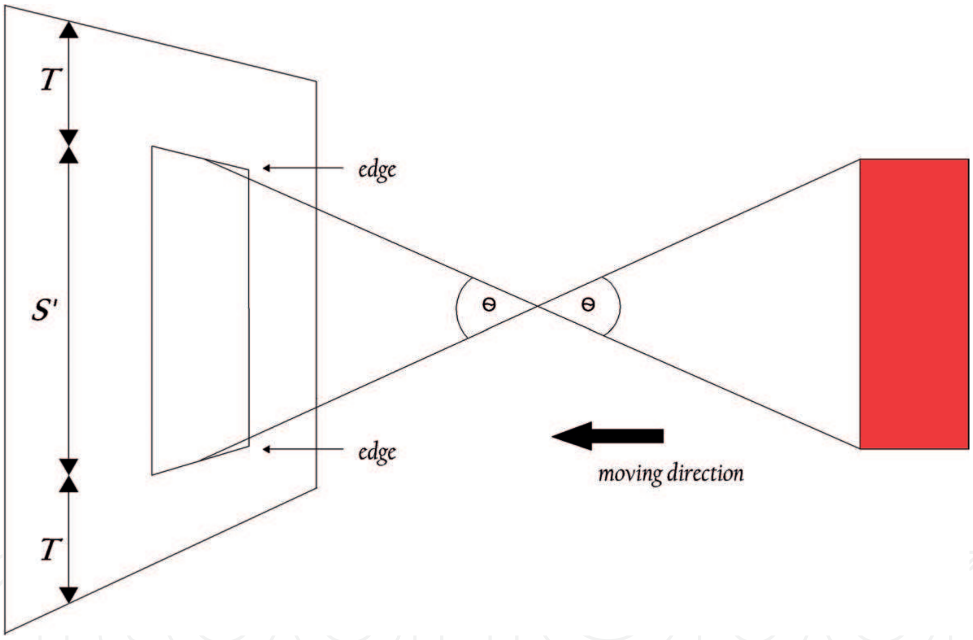

Figure 5. Perspective view of the approaching process.

best case is when the obstacle approaching the camera is alienated with it. In this case, to find the value of θ, the triangle ACD (see **Figure 6**) is divided into two right triangles (ABD and BCD) and from opposite angles by a vertex, we can estimate θ_1 by Eq. (3). Since $\theta_1 = \theta_2$, rearranging Eq. (3) we obtain θ as is shown in Eq. (4)

$$tan(\theta_1) = \frac{\frac{S'}{2}}{f} \tag{3}$$

$$\theta = 2\left[arc\left(\frac{\frac{S'}{2}}{f}\right)\right] \tag{4}$$

3.3. Modeling process

So, suppose we have a mobile robot moving towards an obstacle and we need to be generating a model of the size of the obstacle, that is, as it approaches the obstacle, the camera detects that the obstacle is growing. This is a scenario for the Kalman Filter, because the method is a part of the temporal and tracking models, which when applied to our problem means the tracking of the apparent size of any obstacle over time. The main characteristic of the temporal models is that they relate the state of the system to time $t - 1$ and t as shown by Eq. (5)

$$w_t = \mu_p + Fw_{t-1} + \epsilon \tag{5}$$

In Eq. (4), W_t is an n-dimensional vector of the state components, F is an n-by-n matrix called the transfer matrix or transition matrix, which relates the mean of the state at time t to the state at time $t - 1$, and ε is a random variable (usually called the process noise) associated with random events or forces that directly affect the actual state of the system, and which is normally distributed and determines how closely related the states are at times t and $t - 1$ [16].

In Eq. (4), we can see that it is a recursive model, because we assume that each state depends only upon its predecessor. We assume that W_t is conditionally independent of the states $W_1,\ldots, W_{t-2}$ given its immediate predecessor W_{t-1}, and just model the conditional relationship $\Pr(W_t | W_{t-1})$ [17].

Below, we give a brief description of the specific case Kalman filter. It is not our intention to explain in detail the Kalman filter, we only want to highlight some characteristics and explain the application to our problem.

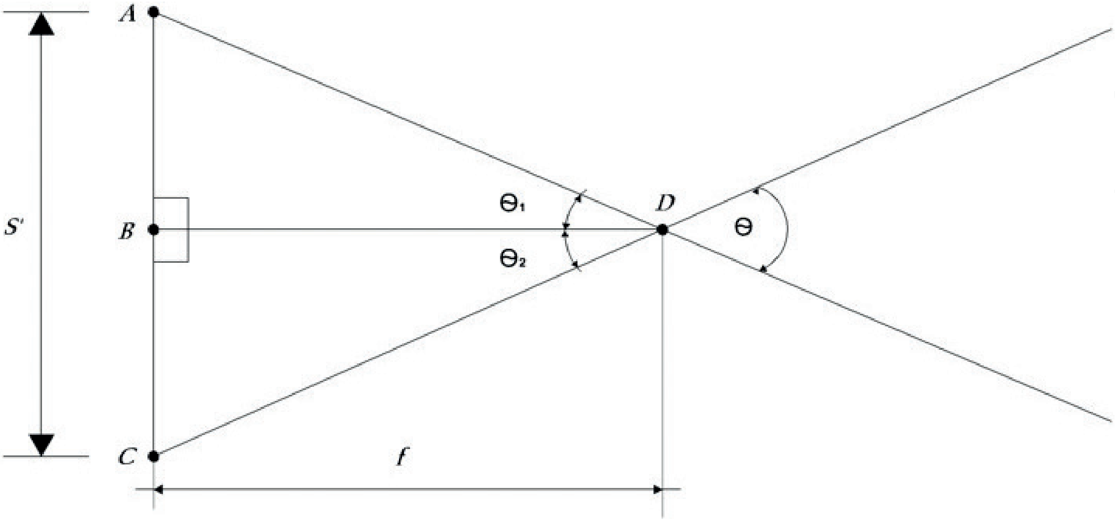

Figure 6. Geometrical view of the camera model.

3.3.1. Description of Kalman filter

The Kalman filter is a set of mathematical equations, described first time in [18], where a recursive solution to the discrete data linear filtering problem is presented. This method has been extensively researched and applied in various fields because it provides us an efficient computational (recursive) mechanism to estimate the state of a process. The filter is powerful because it involves estimations of past, present, and even future states. Kalman filter involves these elements with the use of knowledge of the system and measurement device, the statistical description of the system noises and any available information about initial conditions of the variables of interest [19].

To begin, we must remember that the basic idea of the Kalman Filter is, under a set of assumptions, it will be possible, given a history of measurements of a system, construct a model for the state of the System that maximizes the a posteriori probability of previous measurements. We can maximize *a posteriori* probability without having a long history of measurements above. Instead, we can iteratively update our state model of a system and maintain only that model for the next iteration [16]. These iterations are formed mainly by processes of prediction, measurement, and updating of the state.

Before explaining the process used, we must remember that Kalman Filter is based on three assumptions:

1. The evolution of state space is linear.
2. The errors or noise subject to the measurements are "white."
3. This noise is also Gaussian.

In other words, the first assumption means that the state of the system at time t can be modeled as a matrix multiplied by the state at time $t - 1$. That is good because linear systems are more easily manipulated and practical than nonlinear. The additional assumptions that the noise is white and Gaussian means that the noise is not correlated over time and that its amplitude can be accurately modeled using a mean and a covariance (i.e., noise is fully described by its first and second moments) [16].

3.3.2. Kalman filter applied to TTC

Below, a brief explanation of how the Kalman Filter is used to predict new apparent size measurements of the obstacle to avoid will be given. We emphasize that we do not model the TTC as such, but we model the behavior of the apparent size of the projected obstacle on the image because the TTC depends on this growth.

So, at some time t_1, we determine the apparent size θ to be θ_1. However, because of inherent measuring device inaccuracies (such as changes of light intensity or non-smooth floor mentioned above), the result of the measurements is somewhat uncertain. Then, we decide that the precision is such that the standard deviation involved is σ_1 (only one variable). Thus, we can establish the conditional probability, the value at time t_1, conditioned on the observed value of the measurement θ_1, that is, we have the probability that θ has a value, based upon the measurement we took. At this moment, we best estimate of the $\hat{\theta}_1 = \theta_1$ and the variance $\hat{\sigma}^2_{\theta_1} = \sigma^2_{\theta_1}$.

After, the robot takes another measurement based on color segmentation at time t_2, and therefore, is obtained θ_2 with a variance $\sigma_{\theta 2}$ (which is assumed to be less than the first). To combine these measures, and to obtain a new one with its own variation (Gaussian distribution) Eqs. (6) and (7) are used, where can be seen that the new value is just a weighted combination of the two measured means and the weighting is determined by the relative uncertainties of the two measurements (conditional mean). The weight in these equations can be seen as: if $\sigma_{\theta 1}$ is greater than $\sigma_{\theta 2}$ (that is, more variability), $\sigma_{\theta 2}$ would have more weight because $\sigma_{\theta 2}$ has less variability. Also, the uncertainty in the estimate of new θ has been decreased by combining the two pieces of information [19]

$$\mu = \theta_{12} = \left(\frac{\sigma_{\theta_2}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2}\right)\theta_1 + \left(\frac{\sigma_{\theta_1}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2}\right)\theta_2 \tag{6}$$

$$\sigma_{12}^2 = \frac{\sigma_{\theta_1}^2 \sigma_{\theta_2}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2} \tag{7}$$

Now that we know how to obtain a next measure, we can continue with this process N times (N measurements). This is because we can combine the first two, then the third with the combination of the first two, the fourth with the combination of the first three, and so on [16]. This is what happens when we are tracking the θ over time, we obtain one measure followed by another followed by another.

Usually, Eq. (6) is rewritten as Eq. (8), and Eq. (7) as Eq. (9) because with this new forms, we can separate the old information from de new information. The new information $(\theta_2 - \theta_1)$ is called *innovation*

$$\widehat{\theta_2} = \theta_1 + \frac{\sigma_{\theta_1}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2}(\theta_2 - \theta_1) \tag{8}$$

$$\widehat{\sigma_2^2} = \left(1 - \frac{\sigma_{\theta_1}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2}\right)\sigma_{\theta_1}^2 \tag{9}$$

Finally, Eq. (10) shows our optimal iterative update factor, which is known as the *update gain K*, and so, we obtain the recursive form described in Eqs. (11) and (12). For a more detailed explanation, we suggest to the reader have a look in [19]

$$K = \frac{\sigma_{\theta_1}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2} \tag{10}$$

$$\widehat{\theta_2} = \theta_1 + K(\theta_2 - \theta_1) \tag{11}$$

$$\widehat{\sigma_{\theta_2}^2} = (1 - K)\,\sigma_{\theta_1}^2 \tag{12}$$

Finally, once the behavior has been modeled, it will be possible to forecast TTC by Eq. (2).

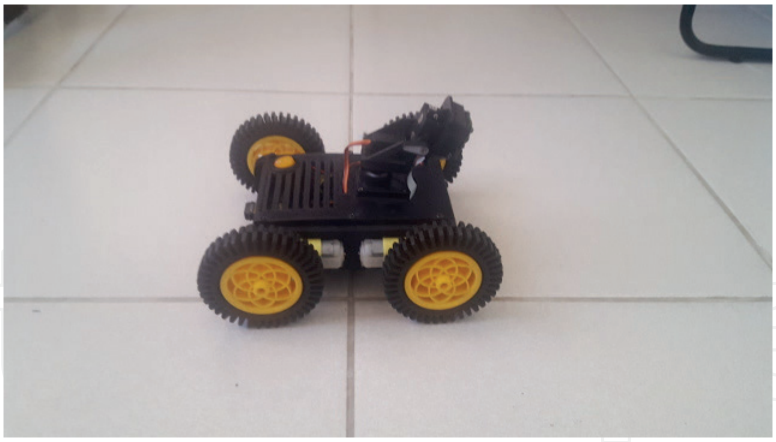

Figure 7. Robot based on Raspberry pi used for experiments.

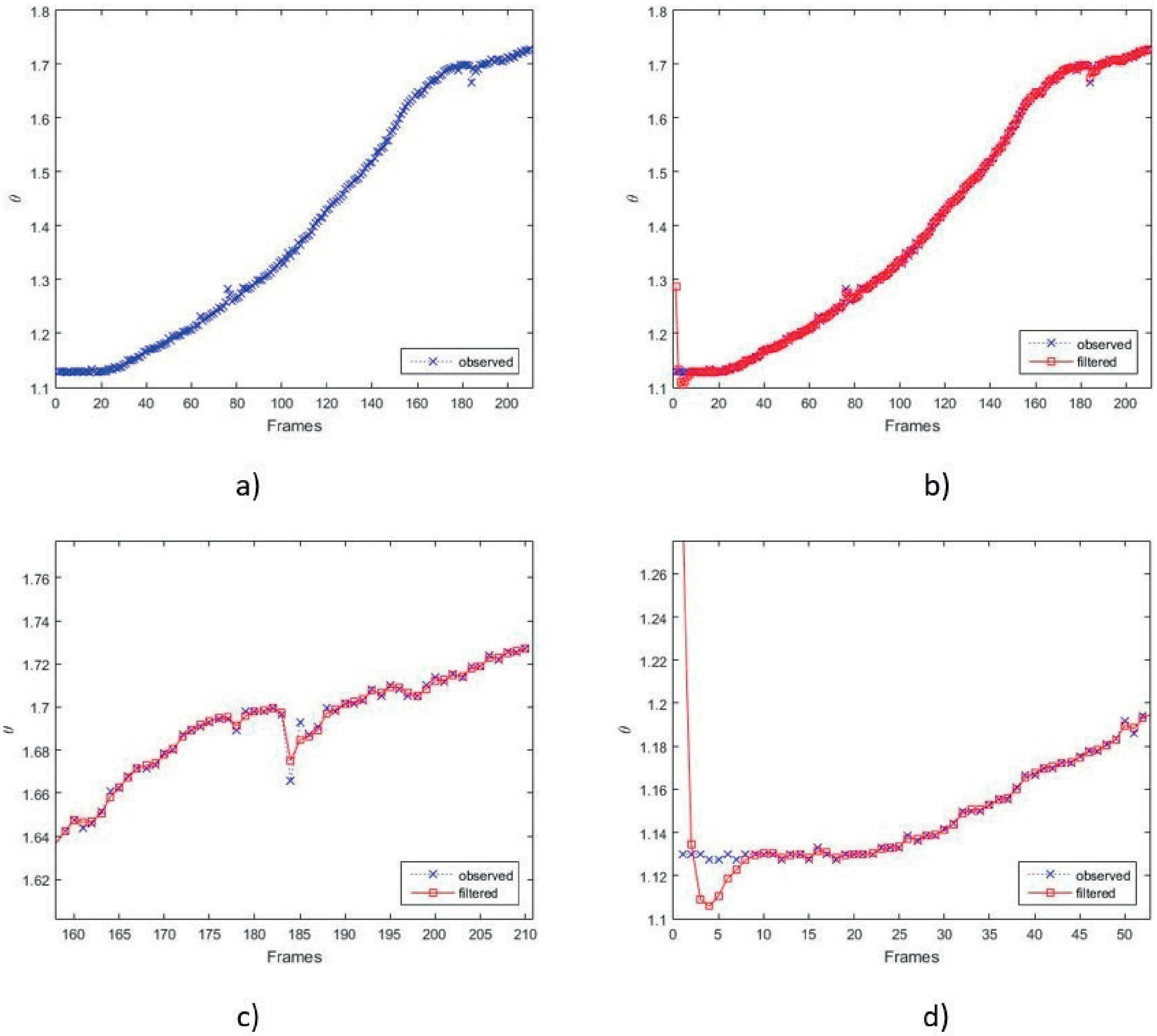

Figure 8. Results of experiments of scenario 1 with a red cylinder. (a) Measurements. (b) Filtered data. (c) Window example, where there is greater variation. (d) Initial convergence process.

4. Experiments and results

In order to test our proposal, we design experiments in which a robot approaches two obstacles mentioned in the segmentation section (see **Figure 3**) at constant velocity and we took 210 frames (equivalent to 7 s) of these real scenarios. A camera was used as a sensor to locate the objects in the robot environment.

Figure 7 shows the robot used for the experiments. In **Figures 8** and **9**, the results of the filtering process are shown for the experiments with the red cylinder and green sphere, respectively. For both figures, (a) shows blue, the measurements obtained by the sensor (specifically by the process of segmentation by color in each image). As can be seen, growth of the θ (obtained by the camera) over time is not "smooth," due to various factors in the environment mentioned above. (b) Indicates red color, the Kalman filtering on the measurements, where it can be seen that they are very close. (c) Shows, for each case, a window of the results where there is greater error in the measurements, which leads to have errors in the prediction but it getting closer to the measurements. (d) Shows an expansion at the beginning of (b), where

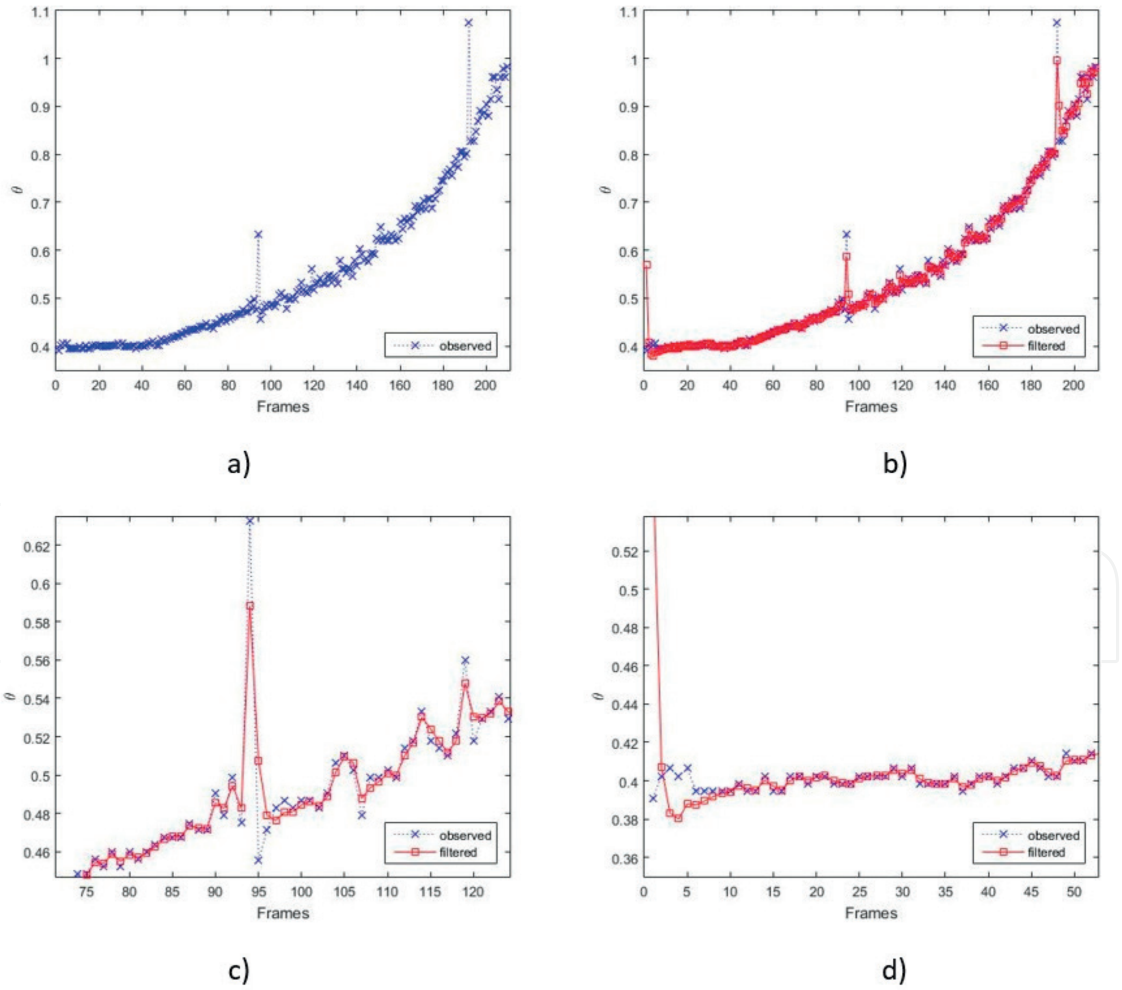

Figure 9. Results of experiments of scenario 2 with a green sphere. (a) Measurements. (b) Filtered data. (c) Window example, where there is greater variation. (d) Initial convergence process.

it can be seen that in both cases it takes little less than 10 frames (about 0.3 s) to converge to obtain estimates close to the measurements.

5. Conclusions and future work

In this chapter, an approach for estimating possible collisions was presented. A brief description of different approaches used to address the TTC problem was analyzed. Taking into account the advantages and disadvantages of these approaches, we present a way to handle the problem by modeling the behavior of the apparent size of the segmented obstacles, which the robot senses in each frame. This apparent size calculation is formally described and used to obtain measures of obstacles. We also apply Kalman filtering as a mechanism to model and predict θ, which will ultimately serve to predict the TTC and therefore avoid collisions. Some features of the Kalman Filter are also highlighted and we describe how it does the estimation for our problem. Finally, our approach is applied to two real cases where the modeling process is observed and the proximity to the measures taken, reducing noise of the measurements. This approach is gaining strength because it is easier to predict (given some measures taken previously) than to be looking for obstacles in each frame. In addition, the Kalman filter will correct errors if measures are incorporated sometimes. As future work, we will continue working on several scenarios and comparing this approach with some others, such as modeling and predicting using system identification techniques or time series.

Author details

Angel Sánchez[1]*, Homero Ríos[1], Gustavo Quintana[2] and Antonio Marín[1]

*Address all correspondence to: angelsg89@hotmail.com

1 Research Center of Artificial Intelligence, University of Veracruz, Mexico

2 ELEC Department, Vrije Universiteit Brussel, Belgium

References

[1] Lee DNA. Theory of visual control of braking based on information about time-to-collision. Perception. 1976;**5**(4):1005020. DOI: 10.1068/p050437

[2] Byoung-Kyun S, Jun-Seok Y, Eok-Gon K, Yang-Keun J, Jong Bum W, Sung-Hyun H. A travelling control of mobile robot based on sonar sensors. In: 15th International Conference of Control, Automation and Systems (ICCAS); October 13-16, 2015; Busan, South Korea. IEEE; 2015. pp. 13-16. DOI: 10.1109/ICCAS.2015.7364821

[3] Vaščák J, Hvizdoš J. Vehicle navigation by fuzzy cognitive maps using sonar and RFID technologies. In: 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI); January 21-23, 2016; Herlany, Slovakia. IEEE; 2016. DOI: 10.1109/SAMI.2016.7422985

[4] Stefan KG, Felix E, Thomas M. A real-time low-power stereo vision engine using semi-global matching. In: Mario Fritz, Bernt Schiele and Justus H. Piater, editors. In: 7th International Conference on Computer Vision Systems; October 13-15, 2009; Liege, Belgium. Springer, Berlin, Heidelberg; 2009. p. 134-143. DOI: 10.1007/978-3-642-04667-4_14

[5] Muffert M, Milbich T, Pfeiffer D, Franke U. May I enter the roundabout? A time-to-contact computation based on stereo-vision. In: Intelligent Vehicles Symposium (IV) 2012; June 3-7, 2012; Alcala de Henares, Spain. IEEE; 2012. pp. 565-570. DOI: 10.1109/IVS.2012.6232178

[6] Benamar F, El Fkihi S, Demonceaux C, Mouaddib E, Aboutajdine D. Gradient-based time to contact on paracatadioptric camera. In: IEEE International Conference on Image Processing, ICIP'2013; September 15-18, 2013; Melbourne, Australia. IEEE; 2013. pp. 5-9. DOI: 10.1109/ICIP.2013.6738002

[7] Alenya G, Negre A, Crowly JL. Time to Contact for Obstacle Avoidance. In: Proceedings of the 4th European Conference on Mobile Robots (ECMR) 09; September 23-25, 2009; Dubrovnik, Croatia. 2009. pp. 19-24

[8] Izumi S, Yamaguchi T. Time-to-contact estimation on the scaled-matching of power spectra. In: SICE, 2007 Annual Conference; September 17-20, 2007; Takamatsu, Japan. IEEE; 2008. pp. 2885-2889. DOI: 10.1109/SICE.2007.4421482

[9] McCarthy C, Barnes N, Mahony R. A robust docking strategy for a mobile robot using flow field divergence. IEEE Transactions on Robotics. 2008;**24**(4). DOI: 10.1109/TRO.2008.926871

[10] McCarthy C, Barnes N. A unified strategy for landing and docking using spherical flow divergence. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012;**34**(5):1024-1031. DOI: 10.1109/TPAMI.2012.27

[11] Horn BKP, Fang Y, Masaki I. Hierarchical framework for direct gradient-based time-to-contact estimation. In: IEEE Intelligent Vehicles Symposium, 2009; June 3-5, 2009; Xian, China. IEEE; 2009. pp. 1394-1400. DOI: 10.1109/IVS.2009.5164489

[12] Horn BKP, Fang Y, Masaki I. Time to contact relative to a planar surface. In: IEEE Intelligent Vehicles Symposium, 2007; June 13-15, 2007, Istanbul, Turkey. IEEE. 2007. pp. 68-74. DOI: 10.1109/IVS.2007.4290093

[13] Kaneta Y, Hagisaka Y, Ito K. Determination of time to contact and application to timing control of mobile robot. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2010; December 14-18, 2010; Tianjin, China. IEEE; 2010. pp. 161-166. DOI: 10.1109/ROBIO.2010.5723320

[14] Sanchez A, Rios H, Marin A, Verdin K, Contreras G. Estimation of time-to-contact from Tau-margin and statistical analysis of behavior. In: International Conference on Systems,

Signals and Image Processing (IWSSIP), 2016; May 23-25, 2016; Bratislava, Slovakia. IEEE; 2016. pp. 1-6. DOI: 10.1109/IWSSIP.2016.7502702

[15] Muller D, Pauli J, Nunn C, Gormer S, Muller-Schneiders S. Time to contact estimation using interest points. In: 12th International Conference on Transportation Systems, 2009. ITSC ’09; October 4-7, 2009; St. Louis, MO, USA. IEEE; 2009. pp. 1-6. DOI: 10.1109/ITSC.2009.5309851

[16] Bradski G, Kaehler A. Learning OpenCV. 1st ed. USA: O’Reilly Media, Inc.; 2008. p. 555

[17] Prince S. Computer Vision: Models, Learning and Inference. USA: Cambridge University Press; 2012. p. 580

[18] Kalman REA. New approach to linear filtering and prediction problems. Transactions of the ASME – Journal of Basic Engineering. 1960;**82**(1):35-45. DOI: 10.1115/1.3662552

[19] Maybeck P. Introduction. In: Stochastic Models, Estimation and Control. 1st ed. London: Academic Press; 1979. p. 1-16