Expert Systems with Applications 38 (2011) 486–494

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An expert fuzzy system for predicting object collisions. Its application for avoiding pedestrian accidents

J.L. Castro, M. Delgado, J. Medina, M.D. Ruiz-Lozano *

School of Computer Science, University of Granada, Department of Computer Science and Artificial Intelligent, C/Periodista Daniel Saucedo Aranda s/n, E-18071 Granada, Spain

ARTICLE INFO

Keywords:
Intelligent surveillance
Fuzzy logic
3D positioning
Collision prediction

ABSTRACT

Collision among moving objects in space is one of the most common risks in daily life. In this context, we have developed an abstract model that allows to detect the presence of risk of future collisions among objects from the video content analysis. Our proposal carries out several stages. First, a camera calibration process calculates the real location of object in scene. Then, we estimate the object speed and their future trajectory in order to predict possible collisions. All the information of the objects is described in an ontology. Using the properties of objects (such as location, speed, trajectories), we have defined a fuzzy rule that permits to identify whether an object is in danger because another could hit him. The use of fuzzy logic results in two points: the collision detection is gradual and the model can be adjusted through membership functions to fuzzy concepts. Furthermore, the proposed model is easily adaptable to any situation and can be applied on various fields. With the aim of testing our proposal, we have focused on pedestrian accidents, a case of special interest since a lot of pedestrians die or are injured in traffic accidents daily. We have developed an application based on our model that is able to predict, in real time, the traffic accidents where a vehicle could run over a pedestrian. The obtained results in the experimental stage show a high performance of the system.

1. Introduction

Collision among moving objects is one of the most common risks in daily life. However, if a collision between two objects was predicted, in many cases, an unpleasant fact could be avoided. This situation is overcoat in traffic accidents, for example, when a vehicle runs over a pedestrian, when two vehicles crash, when an airplane collides with an obstacle or a vehicle at the airport, etc. For this reason, we propose an expert system that uses video content analysis for predicting collisions, detecting objects in danger and warning of this fact in real time with the aim of improving safety in many situations. Our approach is explained in more detail in Section 3 where we describe the inputs and the outputs of the system and the architecture. The architecture is consisted of one '*Translator Module*' where it carries out the 2D–3D conversion process and one '*Processing Unit*', which includes the knowledge ontology (Section 3.2) and the fuzzy reasoner (Section 3.5).

Nowadays, traffic accidents are one of the leading causes of death in Europe and in the majority of developed countries. As a result, it is growing a great interest on pedestrian protection recently, since many accidents injure and kill pedestrians yearly. Due to this fact, for testing our model, we will focus on pedestrian accidents with the aim of increasing safety for pedestrians and drivers. So, we have developed an application that predicts collisions between vehicles and pedestrians. In Section 4, we describe this application and the obtained results in the experimental stage. Finally, in Section 5, you can see the conclusions obtained from this study.

Next, we offer a brief overview of previous work in this research line.

2. Related work

Currently, the security of people in public and private environments is one of the main topics of concern for most governments and institutions. In the last few years, the new technologies have played a very relevant role in current surveillance systems by providing them with robustness and detecting dangerous situations. Besides, artificial intelligence has played a key role in surveillance. A good example of this kind of innovative algorithms is the video content analysis, also known as intelligent video. This method is getting more and more important in the surveillance systems of last generation.

Therefore, the need of dealing with intelligent video arises in order to improve the accuracy of surveillance systems and to reduce the workload of human operators (Buxton, 2003). In the last two decades, the scientific community has proposed algorithms, techniques and models that constitute the second and the third

* Corresponding author.
E-mail address: mdruilo@decsai.ugr.es (M.D. Ruiz-Lozano).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.088

(still in experimental phase) generation of surveillance systems (Valera & Velastin, 2005). Second generation systems combine CCTV technology and IP surveillance with computer vision algorithms and artificial intelligence, while third generation systems are characterized by their inherently distributed nature and the high amount of used surveillance devices. The main aim of this kind of systems is to provide a good scene understanding and a right interaction with the security guard in real time.

Several stages can be distinguished in visual surveillance systems: model and knowledge acquisition of the monitored environment (Li, Ding, & Liu, 2003; Mittal & Paragios, 2004), detection and tracking of moving objects (Behrad, Shahrokni, & Ahmad, 2001; Fejes & Davis, 1999), object classification (Ferryman, Maybank, & Worrall, 2000; Fusier et al., 2007) and behavior analysis (Borg et al., 2005; Stell, 2004). These systems are able to detect a moving object and tracking its trajectory by classifying them and interpreting their interactions with the rest of elements of the scenario.

Most of video-surveillance systems are based on image processing algorithms with the goal of performing concrete functions, such as the detection and tracking of objects. The number of works that pays special attention to the intelligent analysis of object behavior is not high. However, this phase is one of the most useful when carrying out surveillance, since the results obtained can be used by the security guard to do his work. For this reason, our aim is to provide security system with intelligence by generating new knowledge to work with high-level information and to detect more abstract situations.

One of the most prominent application areas for the development of surveillance systems is the traffic monitoring. The majority of this type of systems can be understood as surveillance-based systems for passively monitoring the traffic. In this context, there are applications of three different purposes:

- Works that '*obtain information on different traffic parameters*' (Collins et al., 2000), such as: number of vehicles per unit of time, vehicle classification, average speed and individual speed of each vehicle. Some works, as Tomas and Garcia (2005), have also been carried out for intelligently managing the information shown to the drivers depending on the road conditions.
- *Traffic control for toll purposes or sanctions*. Most of these systems try to manage the traffic flow by controlling the traffic lights, as Mohammadian (2006) and Vallejo, Albusac, Jimenez, Gonzalez, and Moreno (2009). This last work presents a multi-agent system to the control of driver's abnormal behaviors in a crosswalk. These systems present some problems, for example, some of them need a person to detect incorrect behaviors through a monitor.
- Researches whose objective is '*monitoring to detect accidents automatically*', also called AID (Automatic Incident Detection). These focus on finding irregularities, such as stopped traffic, slow traffic and traffic jams. In this context, we can find works as Bo, Qimei, and Fan (2006), Lee, Hellinga, and Saccomanno (2007) Ismail, Sayed, Saunier, and Lim (2009). In Lee et al. (2007), collisions in freeway traffic are detected. Ismail et al. (2009) described an automated video analysis system that can detect and track road users in a traffic scene, and it classifies them as pedestrian and motorized road users; it identifies important events that may lead to pedestrian–vehicle collisions, and it calculates several severity conflict indicators.

In this last sense, and with the aim of solving lives, we propose a system for traffic monitoring that is able to predict risk of traffic accident where a pedestrian is run over by a vehicle. Our system is based on fuzzy logic because we need a progressive response that is calculated quickly in real time. The detection and prevention in real time is stronger point with respect to other proposals.

3. Our approach: model for predicting potential future collisions

In this paper, we develop an expert system able to predict a collision between any two objects. Our system analyzes video information and emits a visual alarm that warns on possible collisions. A particularized representation for a crosswalk is shown in Fig. 1.

The proposed system has been designed to monitor local scenarios. We define a *scenario* as a monitored environment where occur several events. In our case, these events are captured by one or several video cameras. For example, a scenario can be a crosswalk, an airport, a road.

Our approach is based on a fuzzy rule to detect in advance potential future collisions, in a general way. This study is based on calculating the object real positions (3D positioning). In this respect, the use of 2D positions involves some problems, for example, lack of perspective, occlusion of objects. This could lead to false detection of collisions since it does not know the real positions (unknown the third coordinate).

The *input of our system* is the detection and the 2D tracking of objects from cognitive video analysis. We do not intend to make basic process of 'object detection' from video, because a large number of research works have been performed in this stage. Furthermore, the aim of our approach is based on advance aspects of information analysis and alarms detection. For this reason, our system receives annotated video as input.

The *output of our system* is the detection of future collisions among objects. Depending on the situation of the study, we can study pedestrian–vehicle collisions, vehicle–vehicle collisions, obstacle–airplane collisions, etc. When our system predicts the presence of a collision, an alarm will be activated and it will attract the attention of the objects that are in risk, in real time. In addition, we have developed a desktop application, where you can check the status of the system throughout the day. Finally, when an alarm has been enabled, the system offers and stores an explanation of the reasons that triggered the alarm activation.

Fig. 1. Our approach.

3.1. Architecture

Fig. 2 shows the system architecture, which is consisted of one '*Translator Module*' and one '*Processing Unit*'.

3.1.1. Translator

As mentioned above, our system receives as input the outcomes of a knowledge extraction system about object detection and 2D tracking from video. The Translator Module is responsible for turning input events into data that are represented under the conceptual framework defined in our ontology (see Section 3.2).

The Translator is constituted by a *Server* that is listening to input events. It is important to stand out that input information is analyzed and processed by the Translator, so that it is able to obtain new data in higher level. In this way, the Translator provides a geometric procedure that obtains the real position of the objects (3D positioning). The 3D localization is highly valuable, because as we see next, the prediction is based on distance, current speed and time of collision. However, these calculations would be distorted if we maintain the 2D location, for example, the speed calculation is more highly precise in 'meter per second' than in 'pixels per second'. For this reason, we generate new 3D knowledge that will enhance our system to make future decisions.

3.1.2. Processing Unit

In turn, this unit is made up by the following:

1. The **Server** (***S***), which is responsible for receiving the events that are send by the Translator and for updating the 'Scenario Object Warehouse' (creating new objects or updating the existing ones).
2. The **Scenario Object Warehouse** (***SOW***), which shows the different objects that exist on the scenario in real time. All information that is obtained from the input knowledge extraction system along with the new information generated by our system is stored and integrated in this warehouse. A process of mutual exclusion is performed to access the SOW in order to avoid inconsistencies.
3. The **Alarm Detection Module** (***ADM***). In this module, we analyze the presence of danger due to a possible collision. When the SOW is updated, the ADM consider if there are objects in the SOW that modify the degree of belief depending on their features. The degree of belief is a value between 0 and 1, and it represents the alarm level. If the belief degree of situation presence exceeds the threshold, the alarm will be activated.
4. The **Object Eliminator** (***OE***), which is a threat that is released regularly. This process is designed to verify that the objects in SOW are right objects of the real scene, in other words, whether these objects are active objects. So, OE finds objects that have not been updated for some time, these objects will be considered as inactive objects on the scenario, and they will be removed by the Object Eliminator. To perform this process, each object has an associated degree of belief that reflects their activity in the scene. The existence of this procedure is very important because if there is not a good cleaning in the SOW, this fact could lead to detecting false alarms.

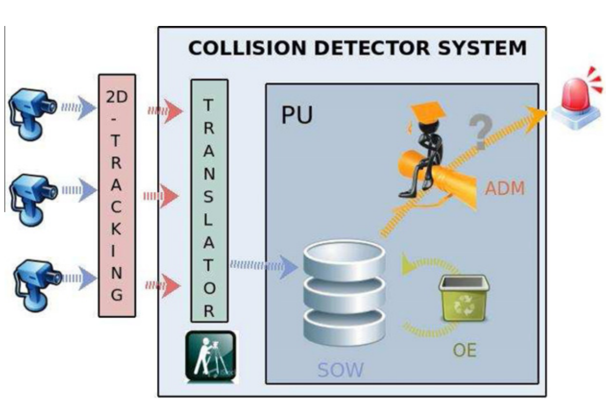

Fig. 2. Architecture.

The communication between architecture modules is implemented using event channels. In this case, we talk about events in the usual sense of the network services. In this context, the information can be sent asynchronously. Customers or receivers can subscribe to a channel and kept waiting for news without making a request. So, event channels provide an efficient method of change information method.

Furthermore, the *User Datagram Protocol (UDP)* (Postel, 1980) is used to transmit the data because it improves the results of *Transmission Control Protocol (TCP)*. This is due to UDP does not offer oriented connections and it does not check data. UDP is an Internet Protocol Suite that sends messages without implicit handshaking dialogues for guaranteeing reliability, ordering or data integrity. Time-sensitive applications often use UDP because dropping packets is preferable to using delayed packets. UDP is located in the transport layer.

3.2. Ontology

We have designed an ontology with the aim of representing all the knowledge in a homogeneous way with independence of the system inputs. The two highlight concepts of our system are *objects* and *alarms*. Both concepts are formally defined next.

3.2.1. System objects

A 'system object' is defined as the set $(i, db, li, t, q, a, loc)$, where:

- i is the object identifier.
- db is a degree of belief, which varies between 0 and 1 and indicates the activity level of the object in the scene.
- li is a list with other possible identifiers of i-object.
- t is the time of last update of the object in the system. The used time unit is the millisecond, which is sufficiently precise for any video rate.
- q represents the object qualities. An *quality* is an attribute or property of an object. It represents as a triple (c, v, d), where c denotes the class or type of quality; v is the quality value; and d is the degree of belief of the quality (between 0 and 1). Example: (''*type*'', ''*vehicle*'', 0.7).
- loc represents the set of object locations within the scenario. A *location* is defined with a set (p, v, s, t, i) where p is the position, v denotes the speed; s represents the object real size; t is the time when the object has this position; and i indicates the increase in time since the last object location.

3.2.2. Alarms

An alarm is represented as a set (i, db, u, t, os, ts) where

- i is the alarm identifier.
- db is the degree of belief of the alarm. It varies between 0 and 1 and indicates the alarm level.
- u is the threshold. It belongs to the interval $[0, 1]$. We consider that an alarm is activated when its degree of belief exceeds this threshold.
- t is the time of last update of the alarm. The used time unit is the millisecond.
- os is a structure that summarizes the explanation of the alarm activation. It consists in a peculiar sequence of system objects.

- *ts* is the explanation of the alarm activation in text format.

3.3. Determination of 3D positioning

In order to create an accurate and precise system, we locate the position of objects in the real world (3D positioning) using a camera calibration process.

There are several mathematical methods that allow to find the 2D coordinates of a projected point on the image plane ($Q' = (x', y')$) from the 3D point in the world ($Q = (x, y, z)$). One of them is based on the use of the view transformation matrix (M), which defines the position and rotation of the view (in our case, the camera). The processing perspective consists of changing the scene coordinate system to another coordinate system centered on the observer (camera). This process has two stages:

- *We move the observer to the origin of coordinates*. This applies a translation of the axes X, Y and Z. Considering (t_x, t_y, t_z) as the camera position regarding the origin, the translation matrix is

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -t_x & -t_y & -t_z & 1 \end{pmatrix}$$

- *We rotate the observer sight line to the Z-axis*. In a three-dimensional system, there are three possible angles that can be applied on the observer. A θ-angle on the Y-axis, a ϕ-angle on the X-axis and a φ-angle on Z-axis. Accordingly, the rotation matrices for each angle are

$$R_\theta = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\varphi = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For that reason, the M-matrix is obtained after applying the translation and rotation processes to the camera: $M = TR_\theta R_\phi R_\varphi$.

Thus, the 2D point is $Q' = QM$, where Q is the point in three dimensions and M is the view transformation matrix.

Our problem is the reverse process. We have the object 2D positions in the image, and we need their 3D positions in the real world. A priori, we know that it is not possible because a component is lost in the image: the z-component (or depth).

On the other hand, we know that the photo was taken at a focal length 'd', which represents the image scale. Therefore, we know that the z-coordinate of point is 'd'. For each 2D point (x, y), we can find infinite possible points that are projected on the same coordinate (x, y, d). These points are defined by $P = (x, y, d) = (2x, 2y, 2d) = \cdots = (\mu x, \mu y, \mu d)$, where μ is any real value.

In this new process, we will undo the previous changes to returning the eye of the observer at its original site and get rid of projections for finding the points in the real world.

$$Q = Q'M^{-1} = Q'R_\varphi^{-1}R_\phi^{-1}R_\theta^{-1}T^{-1}$$

First we must undo the rotations. For example, we have to multiply by the inverse matrix of R_φ to undo the first rotation R_φ. It is not necessary to make a diagonalization process, because we know that the inverse matrix consists of applying the negative angle. As we know the rules, $\sin(-\varphi) = -\sin(\varphi)$ and $\cos(-\varphi) = \cos(\varphi)$, we can solve

$$R_\varphi^{-1} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\phi^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\theta^{-1} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

So, the new point without rotations is $P_r(x0, y0, z0, \alpha) = PR_\varphi^{-1}R_\phi^{-1}R_\theta^{-1}$.

Next, we must undo the translation. To do this, we calculate the inverse matrix of the translation matrix T:

$$T(-tx, -ty, -tz)^{-1} = T(tx, ty, tz) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix}$$

Finally, the new point without rotations or translations is

$$P_{r,t} = (x0 + \alpha \cdot tx, y0 + \alpha \cdot ty, z0 + \alpha \cdot tz, \alpha)$$

We must normalize $P_{r,t}$ for calculating non-homogeneous coordinates, obtaining

$$P = P_{r,t} = \left(tx + \frac{x0}{\alpha}, ty + \frac{y0}{\alpha}, tz + \frac{z0}{\alpha}, 1\right)$$

As we knew, the actual point is depending on α, due to lack of depth. For this reason, it has infinite solutions. To obtain a specific value, we introduce a plane that represents the ground where the image was taken. This plane is defined by a vector $N = (Nx, Ny, Nz)$ and a point $S = (Sx, Sy, Sz)$. And thus, its parametric equation is

$$N = xN_x + yN_y + zN_z - |N \cdot S| = 0$$

If we equate the N-plane equation with the P-point, we can determine a value for α. This value represents the intersection or the common point between the N-plane and the line represented by $P_{(r,t).}$

$$\alpha = \frac{(Nx \cdot x0 + Ny \cdot y0 + Nz \cdot z0 - |N \cdot S|)}{(Nx \cdot tx + Ny \cdot ty + Nz \cdot tz)} = \frac{(|N \cdot P| - |N \cdot S|)}{(N \cdot T)}$$

Finally, if we replace α in P, we obtain the searched 3D point:

$$P(x, y, z) = \left(\frac{(x0 \cdot |N \cdot T|)}{(|N \cdot P| - |N \cdot S|)} + tx, \frac{(y0 \cdot |N \cdot T|)}{(|N \cdot P| - |N \cdot S|)} + ty, \frac{(z0 \cdot |N \cdot T|)}{(|N \cdot P| - |N \cdot S|)} + tz\right)$$

This method allows us to calculate object 3D position from its 2D position. This requires to know (with respect to the coordinate axes that we take as reference) the camera location, the angles of camera inclination on different axes and the focal distance (see Fig. 3). Usually, these parameters can be measured using topographical instruments.

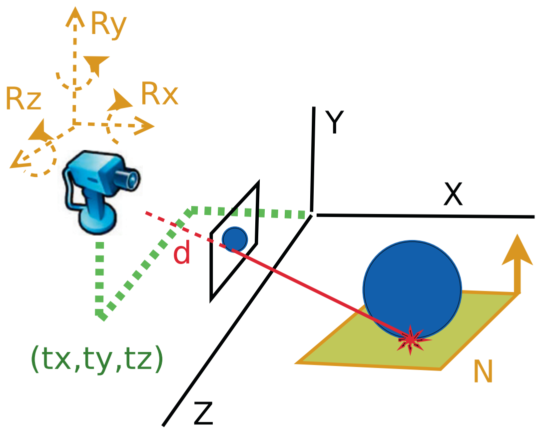

Fig. 3. 2D–3D conversion.

3.4. Estimation of the object speed

The variation in the position of an object in time is defined as the object speed. In our approach, we know the different positions of the objects in time, so that we can obtain the object speeds.

$$S_{current} = \frac{P_{current} - P_{before}}{t_{current} - t_{before}}$$

However, the actual speed can vary over the estimated one, due to the white additive error from image process, the 2D tracking and the 3D conversion. In order to minimize it, we adjust the current speed of each object according to their N previous speeds. So, the calculation of the adjusted speed (S^*) of an object in an instant t_k is

$$S^*_k = \sum_{i=1}^{n}(\alpha_i \cdot S^*_{k-i}), \sum_{i=1}^{n}\alpha_i = 1$$

where α_i are weights that belongs to the interval $[0,1]$.

Thanks to it, possible errors in local calculations are reduced and the abrupt changes in the position are fuzzed.

3.5. Rule for predicting potential future collisions

A priori, to carry out the detection of future collisions, our first proposal was to study the behavior of the mobile objects involved in the collision equally. However, after several tests, we realized that in many situations, it is necessary to distinguish different object roles in a collision: 'vulnerable' or 'threatening'. In this way, our model can differentiate between relevant and irrelevant collisions and take into account only the first ones. For example, if a person runs toward a stopped car, the collision is not relevant; in the other hand, if a vehicle runs toward a person, may be it is. Therefore, we rely on the idea that an *A*-object will collide with a *B*-object. The *A*-object is called 'threatening-object' because it is the cause of the collision. The *B*-object is called 'vulnerable-object' because it is in danger from collision. We have conducted the division of the objects into *A* and *B* roles, because there are situations where an only object is dangerous, for example, the crash between a pedestrian and a car. However, in other situations, both are dangerous, such as two cars that collide at an intersection. In this case, the situation has been resolved by symmetry. First, our model evaluates an object as *A*-role and the other one as *B*-role, and then the opposite situation. So, the two objects are regarded as vulnerable and threatening.

We have solved the prediction of possible future collisions among objects by means of the assessment of a fuzzy rule. The rule is based on three variables:

- *The estimated time for the collision between the 'threatening-object' and the 'vulnerable-object'* ($t_{collision}$).
  Our system obtains the time that is needed by a 'threatening-object' crash with a 'vulnerable-object'. For calculating these data, we need to know the point (*Pc*, *Point of collision*), where the collision will occur. We have defined it as the point that is located in the intersection between the direction vector of the 'threatening-object' ($Pr + \vec{Sr}$, *Position and Speed of 'threatening-object'*) and the Plane of Movement of the 'vulnerable-object' (*Pm*).
  In geometric terms, a plane is defined by a perpendicular vector to it and a point in it. We know that the vectorial product between the vector perpendicular to the scene ground (*Pg*, Plane of ground) and the direction vector of a 'vulnerable-object' ($\vec{Sv}$, Speed of 'vulnerable-object') defines the perpendicular vector to Plane of Movement ($\vec{Pm} = (\vec{Pg} \times \vec{Sv})$). This vector and the position of the 'vulnerable-object' *Pv* allow us to calculate the equation of *Pm* in general way:

$$\vec{Pm}\begin{pmatrix}x\\y\\z\end{pmatrix} + |\vec{Pv} \odot \vec{Pm}| = 0$$

In addition, we can extrapolate the future 'threatening-object' position depending on time:

$$Pr' = Pr + S_r \cdot t_{collision}$$

In order to get the point of collision, we can determine the time $t_{collision}$ that the 'threatening-object' spends from its current position *Pr* with its current speed *Sr* until the intersection with the Plane of Movement, $\vec{Pm}$. So,

$$Pr' \bigcap \vec{Pm} \Rightarrow t_{collision} = \frac{|Pv \cdot Pm| - |Pr \cdot Pm|}{Sr - Pm}$$

The geographical representation of this method is shown in Fig. 4.

- *The distance between the 'vulnerable-object' and point of collision with the 'threatening-object'* (*dist*).
  As we have obtained $t_{collision}$, we can determine the future position where this 'vulnerable-object' and the 'threatening-object' will be located at the moment $t_{collision}$:

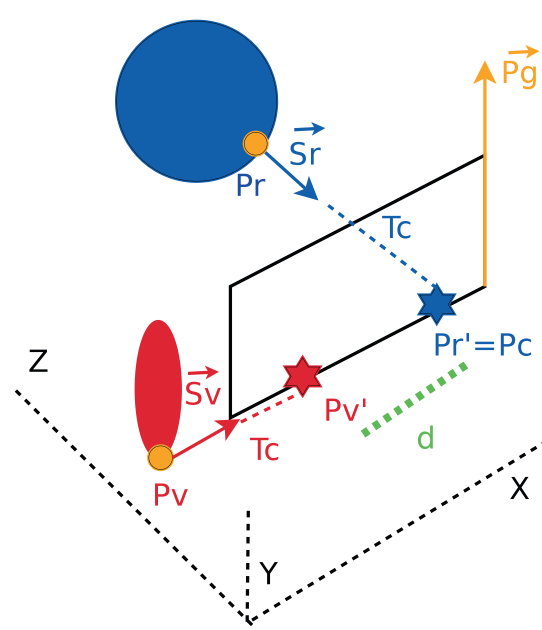

Fig. 4. Process geometry.

$$Pv' = Pv + (Sv \cdot t_{collision})$$
$$Pr' = Pr + (Sr \cdot t_{collision})$$

where Pv and Pr are the 'vulnerable-object' and 'threatening-object' position, and Sv and Sr are the 'vulnerable-object' and 'threatening-object' speed, respectively.
Finally, we calculate the distance '*dist*' between the 'vulnerable-object' and the 'threatening-object':

$$dist = \sqrt{(Pv'^2 - Pr'^2)}$$

- *The speed of 'threatening-object' at the scene* (Sv).
Another important factor in the study of prevention of collisions is speed. For example, when a high probability that two objects may collide (they will become very close) is detected, it is important to know if one of them has stopped (its current speed is zero), because in this case, the alarm level will be decremented. Also, it is very important 'threatening-object' speed: Sr, since it is more probable that a collision happens if the 'threatening-object' runs at high speed.

The rule used to detect the danger of a collision between a 'threatening-object' and a 'vulnerable-object' is defined informally as follows:

"If there exists a 'threatening-object' that can collide with a 'vulnerable-object' in a 'short time' (w0), and this 'vulnerable-object' is 'near' the point of collision with the 'threatening-object' (w1), and the 'threatening-object' runs at 'high speed' (w2), then there exists a risk that the 'threatening-object' hit the 'vulnerable-object'."

The three conditions of the rule are fuzzy. The values w_0, w_1 and w_2 belong to the interval $[0,1]$. Next, it shows as they are evaluated:

- w_0. Once $t_{collision}$ is known, a trapezoidal function is used to obtain the degree of membership (w_0) of $t_{collision}$ with respect to the fuzzy concept 'short time', (see Fig. 5). A trapezoidal function is used because is a good evaluator for human reference in this case. This is due to, first, when the 'threatening-object' overtake the plane of the 'vulnerable-object', the time $t_{collision}$ become negative, so it is not relevant. The degree of membership of obtained negative time with respect to the concept 'short time' is 0. Second, there exists a critical time interval (when there is very short time to collision) where the degree of membership is 1. Third, when $t_{collision}$ is high (a long time to collision), it is not relevant either. So, the degree of membership is 0 again. Between the first and second case, and the second and third cases, we have not considered a discrete change, but gradual.
- w_1. Also, we use other function that indicates the degree of membership (w_1) of *dist* with respect to the fuzzy concept 'near', (see Fig. 5). Let us observe that our approach relies on a reference distance d. With the function used, we assume that if distance between two positions is less than $0.5d$, they are nearby. In contrast, if the distance exceeds $d + 0.5d$, probably they will not be nearby. In intermediate cases, it will be decreasing the certainty that they are close.
- w_2. We use a function that indicates the degree of membership w_2 of the 'threatening-object' speed S_v with respect to the fuzzy concept 'fast', (see Fig. 5).

The evaluation of the fuzzy rule is carried out using the *fuzzy AND operator* over w_0, w_1 and w_2. We have used here the 'minimization function' to represent the *fuzzy AND operator*. As we consider, the alarm level assessment is the lowest of the three values:

$$AlarmLevel = Min(w_0, w_1, w_2), \ AlarmLevel \in [0..1]$$

As we can see, it is necessary that the three conditions (w_0, w_1, w_2) are carried out in order to the risk of collision exists. Alarm level is a degree value that belongs to $[0,1]$ interval. We have divided this interval into subintervals, which we have assigned a label. Thus, the system output (the alarm level) is reinterpreted to determine the risk level in the scenario such as zero, low, medium and high. These states are in function on a threshold value. The threshold of the alarm also belongs to $[0,1]$. Besides, we have associated a color for each alarm-level-state.

- *White state: zero risk*, when the alarm level is zero. No signs of risk of collision among objects.
- *Green state: low risk*, when the alarm level belong to $(0, \text{threshold}/2]$.
- *Yellow state: medium risk*, when the alarm level belong to $(\text{threshold}/2, \text{threshold})$. When the color is yellow, the system alarm emits a warning visual that attracts the attention of involved objects.
- *Red state: high risk*, when the alarm level belong to $(\text{threshold}, 1)$, it is very likely to occur a collision. In this case, the emitted alarm is more intense to stop the involved objects.
The complete process is summarized in Fig. 6.

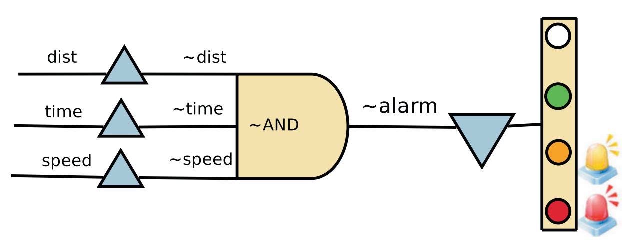

Fig. 6. Fuzzy process.

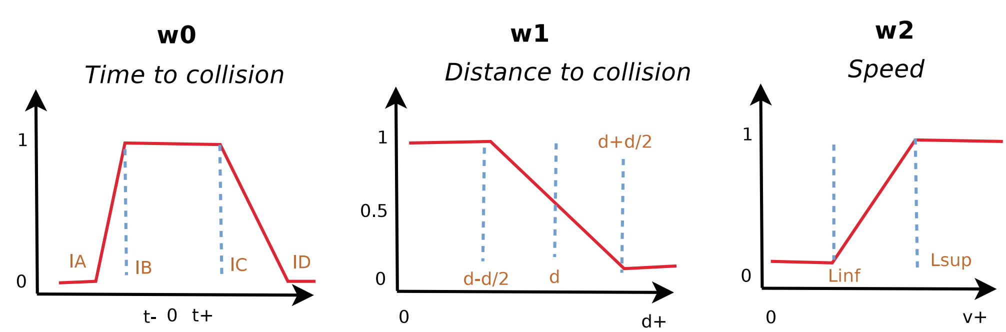

Fig. 5. Memberships to the fuzzy concepts: ($w0$) short time; ($w1$) near; ($w2$) fast.

4. The application for avoiding pedestrian accidents

4.1. Motivation

Daily, a lot of pedestrians die or injured in traffic accidents. Many of these accidents are due to the lack of minimum visibility that prevents drivers for seeing a pedestrian in time to brake, the distance among crosswalks that leads pedestrians to cross outside of them, the lack of a specification that indicates mandatory speed reduction in the presence of a crosswalk, etc. In addition, it is important to emphasize that the great majority of these accidents happen on roads or outside of crosswalks.

According to a study carried out by the foundation RACC (*Real Automobil Club de Catalunya*), we know that the number of pedestrian deaths has decreased in Europe and in Spain in the last 5 years. However, Spain has the highest mortality ratio in Europe with 15.7 deaths per million inhabitants. During the year 2007, in Spain, 591 pedestrians were killed in traffic accidents, both on roads and in urban areas, and 10,838 pedestrians were injured. The reduction in pedestrians killed in urban areas continues at a rate lower than on the road. Other important data are that over 90% of killed pedestrians in Spain were killed outside the crosswalk. Moreover, in Spain, there are deficiencies in the crosswalk design that do not have traffic lights.

In this context, and in order to increase safety, we present a solution that allows to detect the presence of risk of traffic accident where a pedestrian is in danger because a vehicle could hit him. Our goal is *to predict* an accident in time and *to avoid* an unpleasant fact. In order to resolve this, we have adapted the model proposed in the previous section. This new system analyzes the content of video from cameras and detects the presence of future collisions between pedestrians and vehicles.

4.2. Application

We use the general collision prediction model (see Section 3) to solve a concrete situation: to detect risk presence of a pedestrian can be run over by a vehicle. Our approach is based on the idea of identifying in advance a possible collision between an 'vehicle–object' and an 'person–object'. Thus, we have adapted the abstract model to this situation; in this case, the vehicle is the 'threatening-object' and the pedestrian is the 'vulnerable-object' (see Section 3.5).

As mentioned above, our system receives as input the outcomes of a knowledge extraction system about object detection and 2D tracking from Video. So, we have used the results of a work that is described in Silla-Martinez (2008), which carries out detection and 2D tracking of moving objects. This system analyzes video from each camera, frame by frame, and uses the MPEG-7 format to make the video annotation. In each frame, the moving objects are detected and are classified as people or vehicles with a degree of belief. Each detected object is encompassed by an ellipse. The two ellipse radios are known (object size in two dimensions). Furthermore, in the order to indicate the 2D tracking in the annotation, each detected object has an associated list with those objects of the previous frame that are its antecessors. In other words, the detected object is connected with one or several object/s in the previous frame.

In conclusion, the information that we know about each object, using the outcomes of the research described in Silla-Martinez (2008), can be represented by the set (i, p, s, c, a), where

- i is the object *identifier*,
- p is the *i-object 2D position* within the frame,
- s represents the *i-object 2D size* (it is the height and width of the ellipse),
- c denotes the *i-object classification* into vehicles or people with a degree of belief and
- a is a set that contains the *i*-object *antecessors* in the previous frame.

With the aim of reducing the computational complexity, we process all the obtained information of a frame on a single block. In this way, it transmits a single event by each frame. The video events stream that our system currently receives as input is defined by the following set (ci, f, t, ol) where

- ci denotes the *camera identifier*;
- f is the *frame number*; and
- t is *time*.
- ol a *list of detected objects* in the current frame. We know of each object its identifier, its 2D position, its 2D speed (data calculated by our Translator), its 2D size and their classification into people and vehicles with a degree of belief.

The Translator reads the flow of MPEG-7 and converts it into 3D location events that can be integrated in our system (see Fig. 7).

We have designed a desktop application where you can check the status of the studied situation: pedestrian in danger from traffic accident. In this application, the operator sees the development of alarm estate by means of a graphic bar that changes color and size according to the level of alert (see Fig. 8).

4.3. Experimental results

We have tested the system with real situations, and we have analyzed the video that has been captured from a real urban zone. The used scenario has been a crosswalk without traffic lights (see Fig. 9). The first step was to calibrate the used camera to obtain 3D positions of the detected objects from video.

In order to evaluate the system, we have simulated a set of examples that cover many different situations. We need to analyze situations where there is danger of traffic accidents and also situ-

Fig. 7. Translator.

Fig. 8. Desktop application.

Fig. 9. Test scenario.

ations where a pedestrian is hit by a vehicle. For these reasons, we have simulated annotated video events as system inputs.

We have designed 24 canonical examples of situations that can occur in real life in our selected scenario. Among these examples, there are four situations where it is clear that there is no risk of accident, seven situations where this risk is not easy to detect, five situations where pedestrians are run over by vehicles, four situations where vehicles stop in order to pedestrians to cross and four situations where vehicles do not stop and pedestrians has to wait.

The parameters used in the membership functions to the studied fuzzy concepts ($w0$) short time; ($w1$) near; ($w2fast$) (see Fig. 5) to test the system are

- Trapezoidal function that shows the importance of the time that is required until the collision: lA = −1 s, lB = 0 s, lC = 1 s, lD = 2.5 s.
- Function that shows the importance of vehicle speed: lInf = 0 km/h, as 0 value, and lSup = 20 km/h, as 1 value.
- Function that indicates the significance of proximity of person to collision point: d = 4 m as 0.5 value. For this case, 6 m correspond to 0 value and 2 m correspond to 1 value.
- The chosen alarm threshold is 0.75.

The test stage has been carried out in three parts. First, we have tested our system with the 24 examples, and we have scored the obtained results. Second, the same situations have been analyzed by people, who have recorded their observations. Third, we have compared both results: the system ones and the human ones.

Eight people have taken part in the experimental test stage. We shown the simulated videos to these people. They scored, in each example, the degree of presence (white, green, yellow, red – see Section 6) of danger or collision risk between a vehicle and a person. The data recorded by these people are the maximum estimate that they sensed when they saw the videos. We show the data of this study in Fig. 10.

On the other hand, we have executed our system for these 24 situations. Then, we compare our results with human perception. The obtained results are in Fig. 11. If we compare these results with the previous ones, we find 20 coincidences, two overestimates and two underestimates. These results show a high performance of the system.

In addition, we can observe that the two cases where the system underestimate the human prediction are reasonable because

Studied situations	Collision detection in advance				Average Value	Estate
	Choice carried out by the eigh people					
	White	Green	Yellow	Red		
1	7	1			0,04166667	Green
2	6	2			0,08333333	Green
3	8				0	White
4	1	3	3	1	0,5	Yellow
5			5	3	0,79166667	Red
6		1	7		0,625	Yellow
7	3	3	1		0,20833333	Green
8	4	3	1		0,20833333	Green
9				8	1	Red
10				8	1	Red
11				8	1	Red
12				8	1	Red
13	1	3	3	1	0,5	Yellow
14	1	5	2		0,375	Green
15	1	2	3	2	0,58333333	Yellow
16		1	4	2	0,625	Yellow
17	3	2	2	1	0,375	Yellow
18	3	4	1		0,25	Green
19	6	2			0,08333333	Green
20	2	6			0,25	Green
21	7	1			0,04166667	Green
22			6	2	0,75	Red
23				8	1	Red
24		2	5	1	0,625	Yellow

Fig. 10. Human results.

Studied situations	Human	Our System					Real Collision
	Obtained Estate	Obtenited Value	Obtained Estate	coincidences	over-estimated	under-estimated	
1	Green	0,22	Green	1			
2	Green	0	White			1	
3	White	0	White	1			
4	Yellow	0,64	Yellow	1			
5	Red	0,87	Red	1			
6	Yellow	0,47	Yellow	1			
7	Green	0,14	Green	1			
8	Green	0,72	Yellow		1		
9	Red	1	Red	1			X
10	Red	1	Red	1			X
11	Red	1	Red	1			X
12	Red	0,69	Yellow			1	X
13	Yellow	0,65	Yellow	1			
14	Green	0,24	Green	1			
15	Yellow	0,54	Yellow	1			
16	Yellow	0,45	Yellow	1			
17	Yellow	0,65	Yellow	1			
18	Green	0,31	Green	1			
19	Green	0,65	Yellow		1		
20	Green	0,34	Green	1			
21	Green	0,06	Green	1			
22	Red	0,75	Red	1			
23	Red	1	Red	1			X
24	Yellow	0,7	Yellow	1			
				20	2	2	

Fig. 11. Performance system.

the values are from green to white (both, with silent alarm) and from red to yellow (both, with visual alarm being activated).

In turn, the two overestimate cases correspond on yellow to green values. The cause is due to an adjustment of the system that prefers to avoid dangerous situations instead of it ignore the collisions where pedestrians can be injured.

5. Conclusions and future work

Two are the main results in this paper: (1) An abstract model for predicting collisions among objects using video data. (2) A system that predicts when a pedestrian is in danger because he/she could be hit by a vehicle. This system is a concrete implementation of the proposed abstract model.

One advantage is that the used method can apply to any general scenario where there are moving objects, in other words, is extensible to any type of detection of collisions between any type of objects. The only point that changes is the classification of the objects, which are involved in the collision. In addition, the fuzzy functions that are used in this model are easily adjusted according to each studied situation.

Other strong point of our approach is that our system is able to carry out a processing multi-camera. The used algorithms have been designed to work with different views of a scenario and integrate data from different cameras, thanks to calculation of 3D positioning and the homogeneous ontology. This fact requires that the cameras have been calibrated using the same reference axis on the scene.

Thanks to the use of fuzzy logic the output of the model, in this case the detection of a possible future collision is gradual.

With respect to the application, we have proposed a novel intelligent monitoring system that is able to predict the presence of 'risk of vehicle–pedestrian collision' in a monitored environment. It is important to emphasize that our system not only detects, but it predicts. So, the system can anticipate to the accident between vehicles and pedestrians and generates an visual alarm.

The inputs of the system are results from video analysis, which are integrated and processed to generate new knowledge, as 3D positioning. Our approach is based on a mathematical, geometric and fuzzy model. The geometric process calculates complex features such as the speed and the trajectories of the objects and the estimated time for a future vehicle pedestrian collision. In addition, some mathematical operations to minimize tracking errors are necessary (for example, the time weighting).

After a precise geometric process, the system calculates three variables to evaluate the collision: time to crash, distance to collision point and vehicle speed. These variables are converted into degree values by membership functions to the fuzzy concepts short 'time', 'near' and 'fast', respectively. After obtaining and smoothing these variables, it can be studied together to solve the fuzzy rule. Thus, the system is robust against fuzzy information. The alarm assessment follows a model based on a fuzzy rule that allow us detect future possible collisions between vehicles and people.

This system is a good tool for improving the security people in traffic areas, since the system alerts, in real time, to accidents where the pedestrians can be damaged by a vehicle. Currently, this system displays the alarm in a visual way on a computer. However, in a real scene, it should be located in the same place where an audible or visual alarm would alert pedestrians and vehicles that are involved.

This study will continue with the integration of other knowledge sources (sounds detection, sensors detection, etc). In this way, integration of video–audio-sensors analysis make possible the development of surveillance system more powerful because there will exist more available information at the decision-making process. So, we will detect new situations of risk or interest (new alarms), for example, collisions between vehicles. For these purposes, our system is easily scalable. We will create new 'Translators' to introduce the new knowledge sources, and we will develop new Alarms Detection Modules to carry out identification of new situations. In turn, each alarm situation can be analyzed and implemented in a different way because our system design makes it possible. Thus, different techniques can be used to detect situations from the ontology.

Acknowledgments

The presented research has been realized within the project HESPERIA, http://www.proyecto-hesperia.org, financed by the Centre for the Industrial Technological Development (CDTI) across programmes CENIT and for the companies: Indra, Union Fenosa, Tecnobit, Visual-Tools, BrainStorm, SAC and TechnoSafe.

References

Behrad, A., Shahrokni, A., & Ahmad, M. (2001). A robust vision-based moving target detection and tracking system. In *Proceeding of image and vision computing conference*.

Bo, L., Qimei, C., & Fan, G. (2006). Freeway auto-surveillance from traffic video. In *6th international conference on ITS telecommunications proceedings* (pp. 167–170).

Borg, M., Thirde, D., Ferryman, J., Fusier, F., Valentin, V., Bremond, F., et al. (2005). Video surveillance for aircraft activity monitoring. In *Proceedings of IEEE conference on AVSS* (Vol. 1, pp. 16–21).

Buxton, H. (2003). Learning and understanding dynamic scene activity: A review. *Image and Vision Computing, 21*(1), 125–136.

Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., & Tsin, Y., (2000). A system for video surveillance and monitoring. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University.

Fejes, S., & Davis, L. S. (1999). Detection of independent motion using directional motion estimation. *Computer Vision and Image Understanding, 74*(2), 101–120.

Ferryman, J. M., Maybank, S. J., & Worrall, A. D. (2000). Visual surveillance for moving vehicles. *International Journal of Computer Vision, 37*(2), 187–197.

Fusier, F., Valentin, V., Brmond, F., Thonnat, M., Borg, M., Thirde, D., et al. (2007). Video understanding for complex activity recognition. *Machine Vision and Applications, 18*(3), 167–188.

Ismail, K. A., Sayed, T., Saunier, N., & Lim, C. (2009). Automated analysis of pedestrian–vehicle conflicts using video data. In *Transportation Research Board 88th annual meeting*.

Lee, C., Hellinga, B., & Saccomanno, F. (2007). Real-time crash prediction model for application to crash prevention in freeway traffic. *Transportation Research Record. Journal of the Transportation Research Board*, 67–77.

Li, P., Ding, L., & Liu, J. (2003). A video-based traffic information extraction system. In *Proceedings of the IEEE intelligent vehicles symposium*.

Mittal, A., & Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of the IEEE Computer Society conference on computer vision and pattern recognition* (2) (pp. II302–II309).

Mohammadian, M. (2006). Multi-agents systems for intelligent control of traffic signals. In *International conference on computational intelligence for modelling control and automation and international conference on intelligent agents web technologies and international commerce (CIMCA'06)* (pp. 270–276).

Postel, J. (1980). User datagram protocol. Network Information Center RFC 768. RFC from Internet Society.

Silla-Martinez, M. J. (2008). Sistema de videovigilancia inteligente utilizando anotaciones mpeg-7. Master's thesis, Grupo de Procesado de Imagen y Video, Universidad Politecnica de Valencia.

Stell, J. G. (2004). Part and complement: Fundamental concepts in spatial relations. *Annals of Mathematics and Artificial Intelligence, 41*(1), 1–17.

Tomas, V. R., & Garcia, L. A. (2005), A cooperative multiagent system for traffic management and control. In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems* (pp. 52–59).

Valera, M., & Velastin, S. A. (2005). Intelligent distributed surveillance systems: A review. *IEEE Proceedings Vision, Image and Signal Processing, IEE Proceedings, 152*(2), 192–204.

Vallejo, D., Albusac, J., Jimenez, L., Gonzalez, C., & Moreno, J. (2009). A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications*. doi:10.1016/j.eswa.2009.01.034.

Update

Expert Systems With Applications

Volume 38, Issue 8, August 2011, Page 10718

DOI: https://doi.org/10.1016/j.eswa.2011.03.091

Expert Systems with Applications 38 (2011) 10718

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Corrigendum

Corrigendum to "An expert fuzzy system for predicting object collisions. Its application for avoiding pedestrian accidents" [Expert Systems with Applications 38 (2011) 486–494]

J.L. Castro, M. Delgado, J. Medina, M.D. Ruiz-Lozano *

School of Computer Science, University of Granada, Department of Computer Science and Artificial Intelligent, C/Periodista Daniel Saucedo Aranda s/n, E-18071 Granada, Spain

The below references needs to be corrected as below as it was wrong in the published paper. Inaccurate: Ismail et al., 2009: Ismail, K. A., Sayed, T., Saunier, N., & Lim, C. (2009). Automated analysis of pedestrian-vehicle conflicts using video data. In *Transportation Research Board 88th annual meeting*. Corrected reference is: Ismail, K., Sayed, T., Saunier, N., & Lim, C. (2009) Automated analysis of pedestrian-vehicle conflicts using video data. *Transportation Research Record: Journal of the Transportation Research Board*, *2140*, 44–45.

DOI of original article: 10.1016/j.eswa.2010.06.088
* Corresponding author.
E-mail address: mdruilo@decsai.ugr.es (M.D. Ruiz-Lozano).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2011.03.091