Expert Systems with Applications 39 (2012) 9108–9117

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

An expert fuzzy system to detect dangerous circumstances due to children in the traffic areas from the video content analysis

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

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
Video content analysis
Fuzzy logic
Real time
High-level tracking

ABSTRACT

Currently, an important demand for security exists in many fields. In the area of traffic safety, children are vulnerable elements because they may create a dangerous situation easily. Therefore, in this paper, we propose an intelligent surveillance system to detect, in real time, the danger due to the existence of unprotected children in traffic zones. We analyze the behavior of objects from video content analysis. The developed system is based on fuzzy rules for describing semantically the studied risk. The fuzzy logic provides a gradual danger detection. Besides, the model can be adjusted through membership functions to fuzzy concepts. The system is characterized by being highly scalable and flexible. Moreover, it is portable to any environment. We also highlight in our proposal the high-level tracking module developed, which is based on the classification of objects and 3D positioning. This tracking method is robust to errors that 2D tracking can present. The results obtained in the experimental stage show a high system performance.

1. Introduction

The safety of people and infrastructure is highly demanded in today's society. Surveillance systems have been, and are, widely used to maintain security in monitored environments. The new technologies have played a very relevant role in this line. For example, the analysis of video content appears with the aim of improve the accuracy of the systems, providing them with robustness and detecting dangerous situations.

In the last years, a lot of models and systems about Intelligent Video-Surveillance have been published in the academic world (Hu, Tan, Wang, & Maybank, 2004; Valera & Velastin, 2005). In a visual surveillance system, several stages can be distinguished: model and knowledge acquisition of the monitored environment (Mittal & Paragios, 2004), detection and tracking of moving objects (Han, Joo, & Davis, 2007; Mandellos, Keramitsoglou, & Kiranoudis, 2011; Sanchez, Patricio, Garcia, & Molina, 2009; Zhao & Nevatia, 2004), object classification (Collins et al., 2000) and behavior analysis (Cinbis & Sclaroff, 2010; Cucchiara, Prati, & Vezzani, 2007; Sacchi, Regazzoni, & Vernazza, 2001; Sethi & Roy-Chowdhury, 2010). The main aim of the last generation of this kind of systems is to provide a good scene understanding and a right interaction with the security guard in real time.

The scientific community has made great advances in intelligent surveillance. However, many aspects of the intelligent security systems need improvements. Most of the works found in the literature, only cover the first stages (object detection, object tracking, object classification). The number of systems that focus on studying the behavior of objects is smaller. Therefore, there is a great demand for applications that include analysis of the behavior of objects to detect dangers in real time (overall main objective of intelligent surveillance).

Despite the earlier stages of the intelligent surveillance are the most studied, there is still much work to do. The quality of the known data about objects significantly affects the behavior analysis. Thus, currently, the creation of more robust techniques in the early stages of monitoring is demanded in order to obtain valid data with adequate accuracy for describing the reality of the scenes.

There are problems such as lighting changes, occlusion, proximity between objects, etc., which can entail an erroneous detection, an wrong classification or a false tracking of the objects. In Yilmaz, Javed, and Shah (2006) we can read an extensive survey of object tracking methods and also give a brief review of related topics. In the present work, we study the problem of object tracking and present a proposal of tracking based on the real positions of objects in the scene in order to improve 2D-tracking methods.

In the analysis stage of behavior, there are some issues that can be enhanced, for example:

- The identification of complex situations. In the literature, most of intelligent surveillance systems focus on solving situations very simple.

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

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.051

- The development of applications more adaptable. Many systems are designed for specific domains (in particular environments). If the application scenario changes, the system is not valid, or in other cases, the software adaptation process is very difficult (and porting to another environment is unfeasible).
- The creation of scalable systems. Most systems are designed to monitor a specific aspect and are not scalable. In these cases, system architectures do not facilitate that new reasoning modules or new sensors can be added.
- Improve processing times, which in many systems of the literature are too high, although the results obtained are good. In surveillance systems, the identification of risks in real time is a critical requirement.

For these reasons, we aim to create a system to detect anomalies in real time that analyzes complex scenes. The system is highly scalable, since we propose a flexible architecture that makes it possible to add new analysis modules for detecting new dangers. Furthermore, the system has be designed for being applied to different environments.

One of the most prominent application areas for the development of surveillance systems is the traffic monitoring. In this context, there are applications of three different purposes: works that 'obtain information on different traffic parameters' (Collins et al., 2000), works that carry out a 'traffic control for toll purposes or sanctions' (Mohammadian, 2006; Vallejo, Albusac, Jimenez, Gonzalez, & Moreno, 2009) or researches whose objective is 'monitoring to detect accidents automatically' (Bo, Qimei, & Fan, 2006; Castro, Delgado, Medina, & Ruiz-Lozano, 2011; Lee, Hellinga, & Saccomanno, 2007).

In the latter sense, a little studied aspect is the existence of children in traffic areas. Children, which are not under the protection of adults, can move according to their instincts. This fact may cause a hazard in zones where there are moving vehicles. A child may make unwise movements unwittingly, such as run or play near moving vehicles, crossing a road in an improper location at in inconvenient time.

There are environments such as schools, playgrounds or residentials, which have a road sign that warns of danger from the proximity of a place frequented by children. However, there are other urban areas can also be visited by children and have no the abovementioned traffic sign. In this case, the children become vulnerable points that can cause a traffic accident or being victims.

For that reason, the development of a system to identify these events can resolve many situations of risk caused by children in traffic zones. Importantly, in these cases should alert both children and drivers. In this mode, an unpleasant fact could be avoided.

Thus, in this paper we present an intelligent system, which uses information from video analysis to detect danger due to children in traffic zones. In Section 2 is described the proposed system and its architecture. In order to represent all known information about an object, we have created an Ontology, which is defined in Section 3. In Section 4, we present a new object-tracking-model that is based on the classification of objects and 3D positioning.

We have developed a rule-based model, which is described in Section 5, to analyze the existence of danger. This system facilitates us to know when a risk occurs, since an alarm is generated. Next, the system experimental results are showed in Section 6. Finally in Section 7, we present the conclusions and future work.

2. Our approach: definition and architecture

2.1. Definition

In this paper, we describe an expert system able to detect the danger because of children in hazardous traffic areas. The proposed system consists in a expansion of the system presented by the same authors in Castro et al. (2011). In Castro et al. (2011), a system that detects object collisions from video analysis is described.

In the system proposed in this paper, the system of Castro et al. (2011) has been expanded to detect a new alert: 'the danger due to unprotected children in the traffic areas'. To this end we have developed a new and independent reasoning module (a new fuzzy controller). In addition to this, as discussed below, in the present work, two new features are added upon our former work (Castro et al., 2011):

- Object tracking is supplemented and enhanced with a new algorithm for high-level tracking.
- Alarms are notified to mobile devices in real time in a context sensitive way.

Our proposal has been designed to supervise the surveillance of local scenarios. We define a *scenario* as a monitored environment where occurs several events, so that they can be captured by one or several cameras. Any physical environment where there exist adults, children or vehicles can be a scenario of application for the system (for example: a parking, entry/exit of a school, a place, etc.).

We rely on the idea that our system receives annotated video as input. 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. We want to stress that our work is focused on behavior analysis of the observed objects in a monitoring environment. Our aim is focused on information analysis and alarms detection. For this reason, the input of our system is the detection and the 2D-tracking of objects from cognitive video analysis. In this way, we will work with low-level information obtained from video analysis to generate high-level new knowledge.

2.2. Architecture

Fig. 1 shows the system architecture. We highlight the existence of the two new components (with respect to our former work (Castro et al., 2011)): (1) a high-level tracking module (part of the translator) and (2) a new Alarm Detection Module. Grosso modo, the architecture has two main parts: one '*Translator Module*' and one '*Processing Unit*'.

2.2.1. Translator module

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).

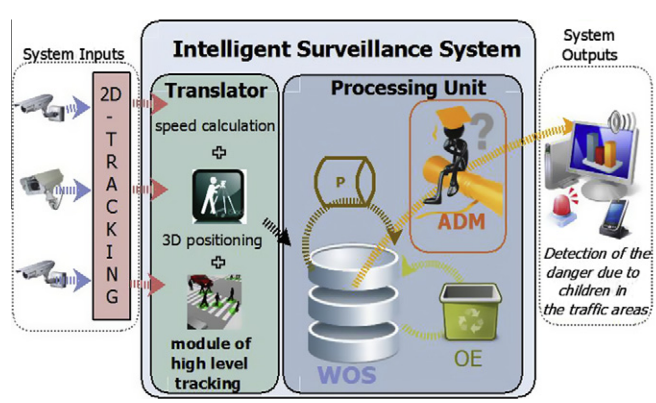

Fig. 1. Architecture.

It is important to stand out that input information is analyzed and processed by the Translator to obtain new data in higher level. In this case, the Translator provides a geometric procedure that obtains the real position of the objects (3D positioning). The estimate of 3D localization is based on a camera calibration process that determines the projection matrix. This method was described in detail by the authors in a previous work (Castro et al., 2011).

Thanks to the new 3D knowledge, we have included in the translator a new process to carry out a high-level objects tracking. This new tracking model is based on the positioning of objects in the real world, their velocities and their classification. This method is described in Section 4.

2.2.2. Processing unit

This unit is made up by:

1. The **Warehouse of the Objects of the Scenario** (***WOS***), 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. This component is updated by the Translator, which creates new Objects or updates the existing ones. A process of mutual exclusion is performed to access the WOS in order to avoid inconsistencies.
2. The **Alarm Detection Modules** (***ADMs***). There exists one ADM by each dangerous situations studied by the system. In this case, we design two ADMs:
   - Module to analyses the existence of danger due to a possible vehicle–pedestrian collision (described in Castro et al. (2011)).
   - Module to analyses the danger due to the existence of children in areas where a vehicle run. This modules is proposed in the present paper and described in Section 5.

   When the WOS is updated, the ADM studies if there are objects in the WOS that modify the degree of belief according to their features. The degree of belief is a value between 0 and 1 that represents the alarm level. If the belief degree of situation detection exceeds the threshold, the alarm will be activated.
3. The **Object Eliminator** (***OE***). This process is designed to verify that the objects in WOS are right and current objects of the real scene. If the 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 OE.
4. A **Plugin** (***P***). Analysis module. Its function is to generate new knowledge from existing information in the WOS. This reasoning module updates the WOS and supplements the information that is known about the objects. This plugin will be defined in more detail in Section 5.

In summary, the *flow of information* of our system is:

1. The system receives as input pre-processed information, which contains data on the number of detected objects, 2D size and position within the image.
2. The *Translator Module* takes the input information and obtains new information at a higher level. Specifically, the Translator obtains the real position of the objects (3D positioning), their speed and carries out a high-level objects tracking that improves results of 2D tracking.
3. All known information of the detected objects is sent to the *Processing Unit*, specifically, to the *WOS* (Warehouse of the Objects of the Scenario). This information is represented using the object schema defined in the Ontology that we propose in Section 3.
4. The ADM (Alarm Detection Module) analyzes information of the WOS and detects the potential dangers or dangerous circumstances due to children in the traffic areas.
5. In parallel, the Object Eliminator (OE) and Plugin (P) analyze the information of the WOS and supplements the information that is known about the objects.
6. The system outputs are the results obtained by the ADM, which generates an alarm indicating the level of danger in areas of traffic when vulnerable children are in the scene. The alarms can be viewed as from a desktop application as from a mobile device.

3. Knowledge representation

We have designed an Ontology with the aim of representing all the knowledge in a homogeneous way with independence of the system inputs. All known information of an object in the scene is represented by the concept 'system Object'. A '**System Object**' is defined as the set (i, db, li, t, q, loc), where:

- **i** is the object identifier.
- **db** is a degree of belief, which indicates the activity level of the object in the scene ($db \in [0, 1]$).
- **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.
- **q** represents the object qualities. A *quality* is a property of an object. It represents as a triple (c, v, d), where: c denotes the type of quality; v is the quality value; d is the degree of belief of the quality ($d \in [0, 1]$). 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; i indicates the increase of time since the last object location.

Other highlight concept of our system is the alarm. 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 indicates the alarm level ($db \in [0, 1]$).
- **u** is the threshold ($db \in [0, 1]$). 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.

4. High-level tracking module

As mentioned above, our system receives as input the outcomes of a knowledge extraction system about object detection and 2D-tracking from video (Silla-Martinez, 2008). Input events provide us with different data on the detected objects in a video frame: the object identifier, its 2D position, its 2D-size, and its classification such as a person, a vehicle or other (with an associated degree of belief). Each object has a different identifier in each frame.

With regard to tracking, each detected object has an associated list of predecessor objects. A predecessor is an object that appears in the previous frame and the tracked object comes from it. A detected object can have one or more predecessors. The ideal situation is each object only has one. However, this situation is not realistic because the object 2D-detection algorithm mixes and confuses the objects. Many models of detection and 2D -tracking objects are sensitive to two situations:

- Situation 1: **Merger of objects**. When the cognitive video analysis detects two or more objects as a single object (due to the problem of occlusion or the proximity between objects). In these circumstances, an object detected in a frame is a fusion of different objects in the previous frame (see Fig. 2).
- Situation 2: **Division of objects**. In this case, several objects detected in a video frame are originated or tracked from the same object of previous frame (see Fig. 2).

These situations present some problems: If an object has several predecessors, what predecessor-object we have to update on the system to keep the tracking? If multiple objects come from a single object of the previous frame, what object keeps the tracking? Is there one that appears the first time? The tracking of one of the objects corresponds to another object that is not his predecessor and does not appear in the previous frame?

Our system must be robust to such situations. For these reasons, in this section, we propose an object tracking algorithm that complements the input 2D-tracking. The new tracking-model is based on the classification of objects and 3D positioning. Its objective is to identify more precise the predecessor for each input frame object. Thus, the tracking of the detected objects in the current frame is associated with existing objects in the system. If there is no possible association, the object is created as a new object in the system.

The *input* of the algorithm is the set of detected objects in the scene in the current video frame. The correct update of the WOS (updating existing objects or creating new objects) is the *output* of the tracking module.

This model uses a matrix *M* and a vector of association *V*:

- *M-matrix*: It is a matrix of $m \times n$, where m is the number of objects detected in the current frame and n the number of detected predecessor objects. Thus, the rows of the matrix refer to the input objects (*SceneObjects*). The columns refer to the objects that are candidates as predecessors of 'SceneObjects (*Candidates*).
  Each cell in the matrix, $M(i,j)$ will contain a value in the interval [0, 1]. This value indicates the degree of belief that the object j is the right predecessor of object i. Thus, for the same row i, the maximum value obtained ($M(i,j)$) indicates that the element j is the best candidate for the detected object i.
  The matrix M is initialized to 0. In the case of that the *Candidates (j)* is not an predecessor of *SceneObjects (i)*, the *M(i*, it j) value is 0, ergo it does not change. But if it is a predecessor, the value $M(i,j)$ is updated. In order to complete the *M*-matrix, we have based on the following idea: *a good predecessor of an object is the candidate-object that shares the same classification (both are two vehicles or two people), and also, is the closest to it*. In this way, we study each pair of objects (i,j):
  - On the one hand, we analyze the qualities of the two objects studied to know if they have the same classification. If so, the degree of compliance with this condition (ω_{class}) is the minimum degree of belief in the qualities that refer to the type of classification of the detected object *i*.
  - On the other hand, in order to know if two objects are close, we study the distance between them. For each candidate object of previous frame, we calculate its position in the current frame (presumably to keep the speed). Subsequently, we calculate the distance between this position and the position of the object studied. Finally, we analyze the degree of closeness between both objects.
    Thus, we use a function that indicates the degree of membership (ω_{class}) of the distance with respect to the fuzzy concept 'near', (see Fig. 3). Let us observe that our approach relies on a reference distance *d*. *d* is a system configuration parameter that must be set by an expert. With the function used, we assume that if distance between two positions is less than 0.5*d*, they are nearby. In contrast, if the distance exceeds *d* + 0.5*d*, probably they will not be nearby. In intermediate cases, it will be decreasing the certainty that they are close.

Therefore, $M(i,j)$ is calculated with the following function:

$$M(i,j) = \omega_{near} \text{ AND } \omega_{class} = min(\omega_{near}, \omega_{class})$$

- *V – association vector*: It is a vector of *m* items. The positions of the vector (i) refer to *SceneObjects*. The content (j) refers to the *Candidates*. The vector reflects the association between the *SceneObjects*(i) with the *Candidates*(j) as the good predecessors.So, we can find the best predecessor to each object calculating the maximum value of the vector $M(i)$.

For each detected object in a frame, the algorithm covers two things:

- it finds the best predecessor (if there are multiple objects predecessors).
- it is able to detach the tracking of two objects in cases where the object of the previous frame is not a good predecessor of the tracked object. This may be due to two cases:
  - The tracked object has no predecessor which shares the same classification and is close to him.
  - In a case of division of objects, a possible predecessor of a particular object, it is best predecessor for another object.

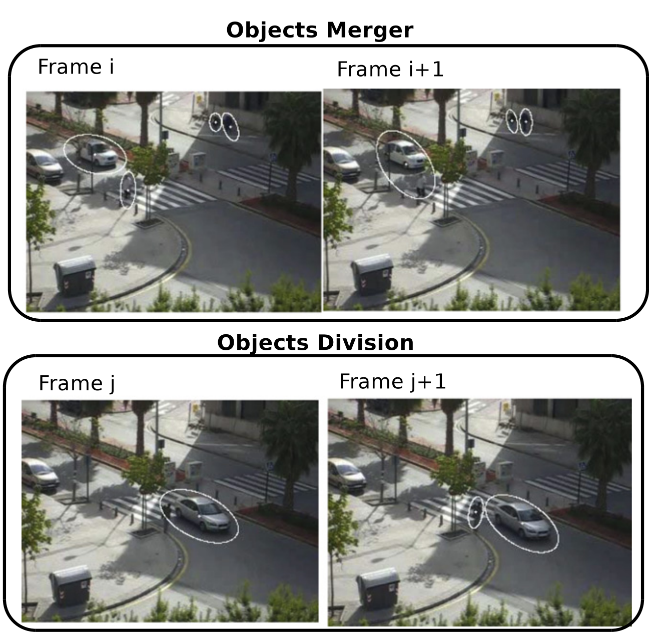

Fig. 2. Example of situations about merging and division of detected objects from video analysis.

In addition, if the tracked object has not a good predecessors, our tracking module checks if there exist another object in the WOS (which does not appears in the previous frame) that can be its predecessor. To do this, we evaluate those objects that share the same classification that the tracked object. And we also analyze whether the object of the WOS, keeping its speed, would be in a position close to the tracked object. If so, we assign the tracking between both objects. Otherwise, we create the object as a new object in the WOS.

This method adequately handles the input information, maintaining consistency and avoiding replicas of objects within the WOS.

The *proposed tracking model* is described by Algorithm 1. The parameters used in Algorithm 1 are: *SceneObjects* is the list of objects detected in current frame. *Candidates* is the list of all possible predecessors. *M* is the *M*-matrix. *V* is the association vector. ω_{class} is the degree of belief about that both objects share the same classification. ω_{near} is the degree of belief about that both objects have nearby positions in the same frame.

Algorithm 1. TrackingModule (*SceneObjects*, *Candidates*)

```
Candidates = {φ}.
for ∀ SceneObjects(i) {i = 0 ... Size(SceneObjects)} do
  Predecessors List = getPredecessors (SceneObjects (i))
  for ∀ object ∈ Predecessors List do
    if object ∉ Candidates then
      Candidates = Candidates ∪ object
    end if
  end for
end for
Create: M[size (SceneObjects)][size (Candidates)], V[size
  (SceneObjects)]
for i = 0 to i = size (SceneObjects) do
  for j = 0 to j = size (Candidates) do
    if Candidates(j) ∉ getPredecessors (SceneObjects (i)) then
      M(i, j) = 0
    else
      ω_class = belief_degree_about _sharing_the_classification
      (SceneObjects(i).classification_qualities(),
      Candidates(j).classification_qualities())
      ω_near = membership_degree_to _tildenear_concept
      (distance (SceneObjects (i), Candidates (j)))
      M(i, j) = ω_class AND ω_near = min (ω_class, ω_near)
    end if
  end for
endfor
while (M ≠ 0) do
  Get maximum value of M and its position
  (Max_Value,i_max,j_max).
if Max_Value ≠ 0 then
    To assig Candidates (j_max) as the best predecessor of
  SceneObjects (i_max): V[i_max] = j_max
    Our system updates 'Candidates(j_max)' with the new
  information of 'SceneObjects(i_max)'
    M[i_max] is updated to 0
  end if
end while
for i = 0 to size (SceneObjects)
  if V[i] = null then
    TocreateNewObject (SceneObjects (i))
  end if
end for
```

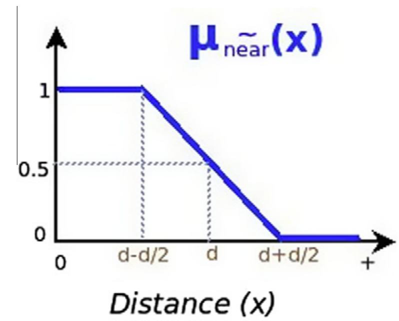

Fig. 3. Membership function to the fuzzy concept 'near.'

5. Module to the detection of risk caused by children in traffic zones

The main objective of our proposal is to develop an expert system to analyze the existence of children in an traffic area generating an alarm in function of the level of danger detected. The system activates an alarm when the behavior of the observed objects implies danger because of children in the traffic areas.

In this study, we focus on the following idea: if there are children in a traffic area unaccompanied by adults, these children are more likely to be able to make imprudent to jeopardize their lives and those of possible drivers. In contrast, if children are accompanied by adults, they can protect, guide and recommend them how to act when there is traffic of vehicles in the area.

Therefore, the model presented here focuses mainly on three aspects:

- The detection of children in the environment.
- The identification of adults who are around or close to children.
- The existence of moving vehicles that are around children.

As we can see, the different aspects to consider using fuzzy concepts $\tilde{near}$, $\tilde{child}$, $\tilde{adult}$. We propose an expert system Based on Fuzzy Rules for studying these concepts.

There are three important parts in the fuzzy control system: a Fuzzifier Module, the Knowledge Framework (the set of objects of WOS and the Rule Base) and the Inference Engine.

5.1. Fuzzifier

The fuzzy concepts that we study in this issue are: $\tilde{child}$, $\tilde{adult}$ and $\tilde{near}$.

We can only rely on the height of the object to determine if a person is a $\tilde{child}$ with the available information. Remember that the 2D-height of an object is an input data of the system. Thanks to the calculation of the position of objects in the real world (3D-position) and 2D-height (h_{2D}) we can estimate the approximate height of an object in the world (h_{3D}):

$$h_{3D} = h_{2D}\frac{d_2}{d_1}$$

where d_2 is the distance between the camera and the object in the world and d_1 is the focal length.

As we can see, we assume that the child concept is defined as people of short stature. Thus, the membership function that defines the *child* fuzzy set is (see Fig. 4):

$$\mu_{\tilde{child}}(x) = \begin{cases} 1 & \text{if} \quad x \leqslant \frac{h}{2} \\ \frac{-1}{h}x + \left(\frac{h+\frac{h}{2}}{h}\right) & \text{if} \quad \frac{h}{2} < x < h + \frac{h}{2} \\ 0 & \text{if} \quad x \geqslant h + \frac{h}{2} \end{cases}$$

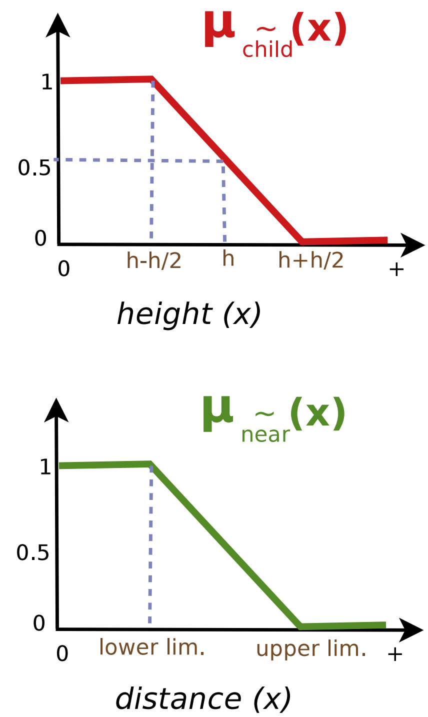

Fig. 4. Membership functions to the fuzzy concepts ‘child’ and ‘near’.

With this function we can determine the degree of belief about if an object is classified as child according to their height. We use a reference value *h*, which indicates an average over the maximum height of children. *h* is a system configuration parameter that must be set by an expert.

For an object classified as a child, with a degree of belief, we use a condition: the object must first be classified as person with a degree of belief greater than a threshold α. In this way, we avoid classifying vehicles as possible children.

The analysis module (Plugin) defined in Section 2.2(architecture) carry out the detection process of children. Therefore, this component will be responsible for classifying people as children using the fuzzification process described above. The plugin will add a new quality (see Section 3) on objects that makes the access to this new information easier. Example: (“*type*”, “*child*”, 0.9).

$\widetilde{Adult}$ is also a fuzzy concept that depends on the height of the object. We define the fuzzy set $\widetilde{adult}$ as the complement of set $\widetilde{child}$

$$\mu_{\widetilde{adult}} = \mu_{\overline{\widetilde{child}}} = 1 - \mu_{\widetilde{child}}$$

$\widetilde{Near}$. Also we use other function that indicates the degree of membership (w_1) of a distance (x) with respect to the fuzzy concept ‘near’, (see Fig. 4). We highlight that our approach relies on a two reference limits $lower_lim$ and $upper_lim$. With the function used, we assume that if distance between two positions is less than $lower_lim$, they are nearby. In contrast, if the distance exceeds $upper_lim$, probably they will not be nearby. In intermediate cases, it will be decreasing the certainty that they are close. These parameters can be adjusted in each studied scenary depending on the proximity among people or between people and vehicles.

$$f(x) = \begin{cases} 1 & \text{if } x \leqslant lim_{lower} \\ \frac{1}{lim_{upper} - lim_{lower}} x + \left(1 - \frac{lim_{upper}}{lim_{upper} - lim_{lower}}\right) & \text{if } lim_{lower} < x < lim_{upper} \\ 0 & \text{if } x \geqslant lim_{upper} \end{cases}$$

5.2. Rule base

The rule base of the fuzzy controller is composed by two rules, which are based on the above ideas. Both rules are defined informally as:

- **Rule 1:** If there is a child on the scenario of study and there are no adults around him, then the alert level is increased. This increase is based on a weight μ_1 and the degree of fulfillment of the antecedent.
- **Rule 2:** If there is a child in the study scenario and there are vehicles that are closely, then the alert level is increased. This increase is a function of μ_2 and the degree of fulfillment of antecedent of this rule.

These rules are evaluated for each child identified in the scene. With the first rule, we study if the nearest adult to the child evaluated is far, which involves a risk. And with the second rule, we check if the nearest vehicle to the child evaluated is near, which entails a danger. On this way, $\forall$ *child_obj* $\in$ *WOS*, we evaluate the rule base:

R1: IF $\nexists$ *person_obj* (*person_obj* $\in$ $\widetilde{adult}$ *AND distancia*(*child_obj*, *person_obj*) $\in$ $\widetilde{near}$), THEN $F_{Increase\_Alarm\_Level}\ (w_{R1}, \alpha_1)$.

R2: IF $\exists$ *vehicle_obj* (*distancia*(*child_obj*, *vehicle_obj*) $\in$ $\widetilde{near}$), THEN $F_{Increase\_Alarm\_Level}\ (w_{R2}, \alpha_2)$.

5.3. Inference engine

The Inference Engine models the human reasoning process, using the rules described above. This engine is evaluated when an new input event is inserted in the system.

We have designed an ad hoc algorithm to evaluate all the rules conditions and to apply the actions defined in the consequents. We have designed an ad hoc algorithm, instead of using a classic motor rules, because, as we discusse below, the need to calculate maximum fuzzy values to evaluate the rules, requires a double cicle of evaluation. The proposed procedure is formally defined in Algorithm 2.

The parameters used in Algorithm 2 are: *WOS* is the set of objects that are in the system. *heigthRef* is the reference height that indicates an average over the maximum height of children. *distRefPeople* is the reference distance to evaluate how close two people are. *distRefVehicles* is the reference distance to evaluate if a vehicle and a child are near. α_1 and α_2 are two weights belong to [0…1], which indicate the increase degree of the alarm level by the rule 1 and 2, respectively. w_{R1} and w_{R2} are the degrees of fulfillment of the antecedents of the rules 1 and 2, respectively. *Alarm_Level* is the level of danger detected after evaluating a specific object, *Max_Alarm_Level* is the global alarm level detected in the scene after evaluating all the objects.

The algorithm examines all objects (one by one) in the WOS. If the analyzed object is classified as $\widetilde{child}$ with a degree of belief greater than a reference threshold, then we evaluate the rule base.

First, we study that there is no adult who is close to the child (evaluation of the first rule):

$$\begin{aligned} \neg(\{&person\_obj_1 \in \widetilde{adult} \wedge distance(child\_obj,\ person\_obj_1) \\ &\in \widetilde{near}\} \ \vee \{person\_obj_2 \\ &\in \widetilde{adult} \wedge distance(child\_obj, person\_obj_2) \\ &\in \widetilde{near}\} \ \vee \ldots \vee \{person\_obj_n \\ &\in \widetilde{adult} \wedge distance(child\_obj,\ person\_obj_n) \in \widetilde{near}\}) \end{aligned}$$

where we apply the minimum function as the operator AND ($\wedge = \wedge_{min}$) and the maximum function as the OR operator ($\vee = \vee_{max}$).

$$(\mu_{\tilde{adult}}(heigth_{person\_obj\_i})) \wedge (\mu_{\tilde{near}}(distance_i)) = wA \wedge wC$$
$$= min(wA, wC) = wAC$$

The degree of fulfillment of the rule antecedent is the complement of the result of the expression in parentheses.

Second, we studied the danger of the vehicle closest to the child (evaluation of the second rule):

$$(distance(child\_obj,\ vehicle\_obj_1)$$
$$\in \tilde{near} \vee distance(child\_obj,\ vehicle\_obj_2)$$
$$\in \tilde{near} \vee \ldots \vee distance(child\_obj,\ vehicle\_obj_n) \in \tilde{near})$$

where we apply the maximum function as the OR operator ($\vee = \vee_{max}$).

Subsequently, we calculate the *local alarm level* to the evaluated child-object. This data is reflected by the fulfillment of any of the two rules. $Alarm\_Level = R1 \vee R2$.

The risk analyzed by the two rules is cumulative. Both rules influence the level of alert. For this reason, we will apply the Lukasievicz method as the OR operator.

$$\vee_{Lukasievicz} = min[1, R1 + R2]$$

After knowing the degree of fulfillment of the two rules (w_{R1}, w_{R2}), we apply the consequent of both:

- For rule 1: $F_{Increase\_Alarm\_Level}(w_{R1}, \alpha_1) = w_{R1} * \alpha_1$.
- For rule 2: $F_{Increase\_Alarm\_Level}(w_{R2}, \alpha_2) = w_{R2} * \alpha_2$.

where α_1 and α_2 are two weights that indicate the degree of increase of both rules respectively. With these weights, we can provide greater or lesser importance to each one of the two main aspects studied.

Consequently, the level of risk identified by the existence of a child in the monitored area is:

$$Alarm\_Level = R1 \vee_{Lukasievicz} R2 = min[1, w_{R1} * \alpha_1 + w_{R2} * \alpha_2]$$

This model produces an alarm level for each child-object identified in the scene. The global alert level detected will the greatest danger to all children identified, in others words, the maximum of each local alarm levels. Even so, this model draws up an explaining about the reasons of danger of all children who are affected. When the controller evaluates a rule, if the degree of fulfillment of the antecedent is greater than a reference threshold, then the explanation is saved. This explanation is issued in two ways:

- In text mode thanks to we know the semantics of the rule.
- In a list of objects represented with the conceptual schema defined in the Ontology (see Section 3).

Algorithm 2. ChildrenDanger_ADM (*WOS*, α_1, α_2, *heigthRef*, *distRefPeople*, *distRefVehicles*)

```
Require: WOS, heigthRef, distRefPeople, distRefVehicles, α1, α2
  Max_Alarm_Level = 0
  for ∀ obj ∈ WOS do
    if ¬g is Child (obj, heigthRef)
      continue
    else
      wR1 = 0
      wR2 = 0
      for ∀ other_obj ∈ {WOS − obj}
        if isPerson(other_obj) then do
          wA = degree_of_belief_Adult (other_obj)
          distP = getDistance (obj, other_obj)
          wC = degree_of_belief_Closeness (distP, distRefPeople)
          wAC = AND (wA, wC)
          wACmax = OR (wR1, wAC)
        else
          if isVehicle(other_obj) then
            distV = getDistance (obj, other_obj)
            wVC = degree_of_belief_Closeness (distV, distRefVehicles)
            wR2=OR (wR2, wVC)
          end if
        end if
      end for
      wR1 = (1 − wACmax)
      if wR1 ⩾ γ1 then
        save_explicationR1 (obj, other_obj, wR1)
      end if
      if wR2 ⩾ γ2 then
        save_explicationR2 (obj, other_obj, wR2)
      end if
      Alarm_Level = min[1, wR1*α1 + wR2 * α2]
      Max_Alarm_Level = max (Max_Alarm_Level, Alarm_Level)
    end if
  end for
```

This alarm level is a degree value that belongs to interval [0, 1]. We have divided this interval into subintervals, which we have assigned a label depending on the danger. Thus, the Alarm Level determines the risk level in the scenario such as zero, low, medium and high. These states depend on a threshold value, which is a parameter that can be adjusted in the system. 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.
- *Green state: low risk,* when the alarm level belong to $(0, \text{threshold}/2]$. There exists a small possibility that there is danger.
- *Yellow state: medium risk,* when the alarm level belong to $(\text{threshold}/2, \text{threshold})$. There is some vulnerable children. When the color is yellow, the system emits a warning sound that attracts the human operator.
- *Red state: high risk,* when the alarm level belong to [threshold, 1]. There exists a significant danger. In this case, the emitted alarm sound is more intense.

We have designed a desktop application where you can check the status of danger detection. In this application, the operator sees the development of alarm state by means of a graphic bar that changes the color and the size according to the level of alert (see Fig. 5). Besides, there exists other application ad hoc mobile devices where the risk alarm level is notified and is also showed by means of a graphic bar. The mobile application have been developed on *JavaME*, using the configuration CLDC (*Connected Limited Device Configuration*) and it can be installed on any mobile phone, even on PDAs, with connectivity, such as Wi-Fi or GPRS.

6. Experimental results

The proposed system has been implemented and tested satisfactorily. The test scenario is a point of entrance and exit of vehicles to a garage (see Fig. 6). In order to evaluate the system, we have simulated a set of 20 different situations that may occur in the selected scenario. To do this, we simulated annotated video

Fig. 5. Desktop application.

events as system inputs. The number of objects in these examples varies between 3 and 7.

The system shows high performance because of the use of fuzzy logic. The logic of the system based on fuzzy rules proposed makes it possible to identify the danger because of being in front of children, which are away from adults or close to vehicles in the scenario under surveillance.

One of our goals is the detection of dangerous situations in real time. For this reason, we have conducted a study about the processing time of the system. To do this, we used the previous proof examples. The computer used is a Pentium (R) Dual-Core CPU E5200@2.50 GHz 2.51 GHz, 2,75 GB de RAM. We have carried out three experiments.

- On the one hand, we have studied the time that the Translator used in process the input data. We measured the time elapsed since the system receives an input event until the Translator updates the WOS. In this phase, several processes are carried out: calculation of 3D positioning, obtaining speed and application of the algorithm for high-level tracking.
- On the other hand, we have evaluated the time that the system used to analyze the behavior of objects in order to detect the danger studied. In other words, the time used by the ADM.

In Table 1 you can see the times (measured in milliseconds) obtained in these experiments. The average times show that the system is efficient. We can see that the ADM processing time is shorter than the time of the Translator. This is because the processes carried out by the Translator have a higher computational cost, such as calculating the 3D position, calculating of the speed and the application of the algorithm for high-level tracking. Still, we are talking about a few hundreds of milliseconds.

Fig. 6. Test scenario.

The maximum time spent by the system in both processes is less than half a second. So, we can say that, in the worst case, the results are generated in real time. In this case, there is only a tiny delay, which is insignificant.

- The third experiment consists of evaluating the system in complex scenes, where the number of objects is high. In this case, we have generated sets of random objects in the WOS and we have run the proposed ADM. In each iteration, the number of

Table 1
Processing times of the system (in milliseconds).

Times	Traductor	Danger detection
Maximum	327	51
Minimum	2	3
Average	140.31	17.51
Standard deviation	61.65	10.66

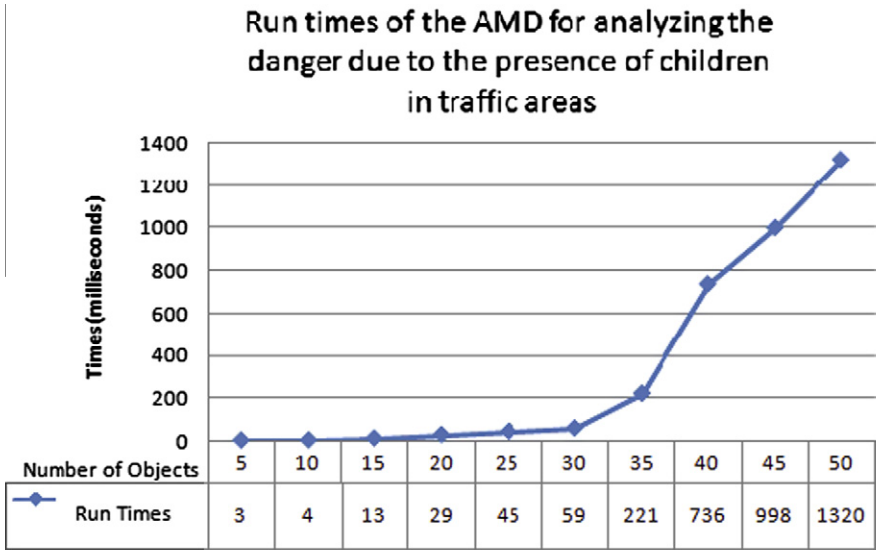

Fig. 7. Run times of module of analysis for the alarm detection.

objects is evaluated by the system to observe increased processing times. The objects are created with a probability of 30% are vehicles, other 30% are people and the other 40% are children. The results of this test are shown in Fig. 7. We note that in scenes where the number of objects is less than 40, the time is less than one second, which is almost negligible. A greater number of objects involved a longer processing. We see that with 50 objects, the time spent is more than a second. The presented system is suitable as a real-time application.

The high-level tracking module has been independently tested. For testing, we used annotated video real results (processed by the system described in Silla-Martinez (2008)). Specifically, we have focused in scenes collected by a camera on a real scenario for 6 min. Our system has processed the information of 2880 frames. The tracking module proposed here (based on the classification of objects and 3D positioning) improve the results of 2D tracking (see Table 2).

The process about detection and the 2D-tracking of objects used as inputs in the system detects more objects than it actually is. This occurs because when two objects come close, both are detected as a single object. And when they are divided, this tracking is not kept. In this case, one of them is identified as a new object in the scene. Our model keeps track in these situations. In this way, we avoid, in most cases, create new objects when they are objects that already exist in the scene. This is very important for later data analysis.

7. Conclusions and future work

In this paper, we proposed an intelligent surveillance system that detects danger in areas of traffic when vulnerable children are in the scene. Thus, this system is a good tool for improving the security of the children and the drivers. The system alerts in real time about the detection of this risk in a concrete zone. Apart from reporting alarms to clients through desktop application, where the operator can check the existence of the situation, the alarms will also be notified via mobile devices. In this way, the operator can be informed without being in front of the command post. Besides, when a hazard situation is detected, the system offers an explication about the events involved in this detection.

Table 2
Results about the number of detected object.

	Real scene	System inputs	Proposed tracking
#People	17	27	19
#Vehicles	16	20	14
#Other	0	5	2
#Total	33	52	35

In order to analyze the existence of the danger due to children in the traffic areas from the video content analysis, we propose a model based on fuzzy rules. This type of system is easily scalable. The analysis of information may be supplemented by means of adding more rules when necessary. This utilization of linguistic variables, fuzzy control rules, and approximate reasoning provides a means to incorporate human expert experience in designing the controller. Moreover, this system has been designed to be robust to fuzzy information. Through the use of fuzzy logic the model output (the detection of risk) is gradual and efficient.

The model is sensitive to the accuracy of data input. The quality of input data influences notably on the system output. For example, the children identification depends on the 2D size that is identified in the detecting and tracking of objects. When there is a considerable distance from the camera to the scene, a few pixels (in the image) may be several decimeters in reality. In most developed currently tracking the size of objects is not accurately detected, it is sometimes overestimated and others underestimated. These errors in the bottom layer are transmitted to the upper layer, where is carried out the process of identification of children. If there is an overestimation or underestimation, then identification of children may have false negatives or false positives. In these cases, children are identified as adults, or adults classified as children, respectively. Therefore, these errors are derived from tracking used as system input and not the process presented previously.

One advantage is that the used method can apply to any general scenario where there are moving objects. In addition, the fuzzy functions that are used in this Model are easily adjusted according to each studied environment.

Other strong point of our approach is the new High-level Objects Tracking Model. We have proposed a solution to various problems tracking 2D models can present. The new tracking-model proposed is based on the classification of objects (such as people and vehicles) and 3D positioning. Its objective is to identify the more precise predecessor for each object detected in a frame. It is robust to the merger situations (several separate objects are detected as a single object when they are close together) or division of objects (an object is divided into several objects, which come from the initial object). Thus, this method properly manages the information received from video analysis, maintaining consistency and avoiding replicas objects within a scene.

Besides, our system is able to carry out a processing multicamera. The used algorithms have been designed to work with different views of a scenario and to integrate data from different cameras, thanks to calculation of 3D positioning and the homogeneous Ontology. Multicamera scenaries require that all cameras have been calibrated using the same reference axis on the scene.

An important advantage of the system is its component-based architecture. This design makes it possible the reuse of components, since they are independent units of development and have multiple uses in different security systems.

The proposed architecture lets the system is easily scalable and flexible. 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.

This study will continue with the integration of other knowledge sources, for example, signaling traffic lights and other sensors, which will allow us to develop new Alarms Detection Modules to detect new risk situations.

Acknowledgments

The presented research has been realized within the HESPERIA project, 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.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2012.02.051.

References

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

Castro, J. L., Delgado, M., Medina, J., & Ruiz-Lozano, M. D. (2011). An expert fuzzy system for predicting object collisions. Its application for avoiding pedestrian accidents. *Expert Systems with Applications, 38*(1), 486–494.

Cinbis, N. I., & Sclaroff, S. (2010). *Object, scene and actions: Combining multiple features for human action recognition* (pp. I: 494–507).

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

Cucchiara, R., Prati, A., & Vezzani, R. (2007). A multicamera vision system for fall detection and alarm generation. *Expert Systems, 24*(5), 334–345.

Han, B., Joo, S., & Davis, L. (2007). Probabilistic fusion tracking using mixture kernel-based bayesian filtering. In *ICCV*.

Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics, 34*(3), 334–352.

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.

Mandellos, N. A., Keramitsoglou, I., & Kiranoudis, C. T. (2011). A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications, 38*(3), 1619–1631.

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* (Vol. 2, pp. 302–309).

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).

Sacchi, C., Regazzoni, C. S., & Vernazza, G. (2001). A neural networkbased image processing system for detection of vandal acts in unmanned railway environments. In *Proceedings 11th international conference on IEEE image analysis and processing* (pp. 529–534).

Sanchez, A. M., Patricio, M. A., Garcia, J., & Molina, J. M. (2009). A context model and reasoning system to improve object tracking in complex scenarios. *Expert Systems with Applications, 36*(8), 10995–11005.

Sethi, R. J., & Roy-Chowdhury, A. K. (2010). Modeling and recognition of complex multiperson interactions in video. *ACM Multimedia MPVA*.

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.

Valera, M., & Velastin, S. A. (2005). Intelligent distributed surveillance systems: A review. *IEEE Proceedings Vision, Image and Signal Processing, 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, 36*(7), 10503–10511. doi:10.1016/j.eswa.2009.01.034.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys, 38*(4).

Zhao, T., & Nevatia, R. (2004). Tracking multiple humans in crowded environment. In *CVPR* (pp. 406–413).