



Contents lists available at ScienceDirect

Expert Systems with Applications

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

OCULUS surveillance system: Fuzzy on-line speed analysis from 2D images

J. Albusac^{a,*}, D. Vallejo^b, J.J. Castro-Schez^b, L. Jimenez-Linares^b

^aSchool of Technical Engineering, Plaza Manuel Meca 1, 13400 Almaden, Ciudad Real, Oreto Research Group, University of Castilla-la Mancha, Spain

^bSchool of Computer Science, Department of Information Technologies and Systems, Paseo de la Universidad 4, 13071 Ciudad Real, Oreto Research Group, University of Castilla-la Mancha, Spain

ARTICLE INFO

Keywords:

Behaviour analysis
Speed classification
Anomaly detection
Surveillance system
Machine learning

ABSTRACT

This paper presents an independent component integrated into a global surveillance system named as OCULUS. The aim of this component is to classify the speed of moving objects as normal or abnormal in order to detect anomalous events, taking into account the object class and spatio-temporal information such as locations and movements. The proposed component analyses the speed of the detected objects in real-time without needing several cameras, a 3D representation of the environment, or the estimation of precise values. Unlike other works, the proposed method does not require knowing the camera parameters previously (e.g. height, angle, zoom level, etc.). The knowledge used by this component is automatically acquired by means of a learning algorithm that generates a set of highly interpretable fuzzy rules. The experimental results demonstrate that the proposed method is accurate, robust and provides a real-time analysis.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In the last two decades, the new technologies have greatly influenced the design of surveillance systems deployed in security control centers. As time passes, these systems are more and more robust when monitoring tasks, reducing the human workload and avoiding dangerous situations. Computer Vision (Forsyth & Ponce, 2002) and Artificial Intelligence (Russell & Norvig, 2003) techniques are playing a key role in the evolution of this kind of systems (Blauensteiner & Kampel, 2004; Bloisi, Iocchi, Remagnino, & Monekosso, in press; Haritaoglu, Harwood, & Davis, 2000; Valera & Velastin, 2005; Velastin, Khoudour, Lo, Sun, & Vicencio-Silva, 2004). A good example can be found in the video analysis field, where one of the main challenges is to develop security expert systems with the autonomy and ability required to automatically understand events and behaviours in order to improve the productivity and effectiveness of surveillance tasks.

In complex environments where multiple situations take place simultaneously, human agent operators have to deal with all of them, being affected by negative factors such as fatigue or tiredness after a prolonged period of observation (Smith, 2004). Nevertheless, artificial expert systems do not have these limitations due to their processing capabilities. Furthermore, this kind of systems can be more effective than people when recognising certain classes of events, such as the detection of suspicious or unattended objects (Dee & Velastin, 2008).

Anomaly detection in real environments may imply the analysis of multiple factors such as trajectory, speed, location, and spatial relationships among objects. In this work, we face the problem of speed analysis from 2D images captured by surveillance cameras. The speed of an object is an important aspect to take into account because a high speed usually represents an abnormal situation. For instance, a vehicle quickly leaving a particular area may be an indication that something is going wrong, or a person running through an area where people normally walk may also be a problem to be detected.

The speed of an object at a particular environment is considered as normal whether its value belongs to the interval in which the allowed limits for that environment have been defined. These limits depend on the characteristics of the environment to be analysed, and they can be upper and lower limits. An object normally behaves according to the speed aspect, whether it does not exceed the upper limit and moves at a speed greater than the lower one. Both kind of limits are not always present in all environments. For example, a lower limit is not common in urban areas where people and vehicles can be stopped without involving an abnormal situation. However, this kind of limit exists in highways where vehicles must overtake a minimum speed.

Many authors have addressed the analysis of speed from 2D images. An easy alternative consists on using aerial views to study the objects displacements in a concrete time interval (Liu, Yamazaki, & Maruyama, 2007). However, it is often difficult to get this kind of images. A second alternative is focused on building a 3D representation of the environment from several cameras in order to know the exact position and movement of each moving object (Cheung, Kanade, Bouguet, & Holler, 2000; Hu, Wang, & Uchimura,

* Corresponding author. Tel.: +34 617342824; fax: +34 926 29 53 54.

E-mail address: JavierAlonso.Albusac@uclm.es (J. Albusac).

URL: <http://www.esi.uclm.es/www/jalbusac/> (J. Albusac).

2008; Lee, Romano, & Stein, 2000). Many of these methods are computationally expensive and the use of several cameras is an expensive solution. Other authors have proposed methods to overcome the same problem by using a single camera (Beymer, McLachlan, Coifman, & Malik, 1997; Cathey & Dailey, 2005; Maduro, Batista, Peixoto, & Batista, 2008; Palaio, Maduro, Batista, & Batista, 2009). In this kind of methods, a calibration process is normally carried out in which parameters such as height, zoom level or angle camera must be previously known. Afterwards, the correspondence between the location of each object in the 2D image and its location in the real world is established.

In this work, we are not interested in measuring the exact speed of each detected object, but to provide a fast response when distinguishing between normal and anomalous speeds in the same way that humans do (Zadeh, 2001). A person does not need to know precise values or perform complex calculations when monitoring a concrete scene in order to determine whether an object is moving fast or slow. In this case, the displacements in the 2D image are analysed by taking into account the perspective of the camera and the static elements of the environment as reference. Medium displacements made by objects in areas which are far away from the camera may represent a high speed. On the contrary, the same displacements in areas close to the camera may involve a normal speed. In the next sections, we propose fuzzy methods (Zadeh, 1965) to learn and analyse the normal displacements of moving objects in 2D images, considering the perspective of a single camera and without previously knowing the camera parameters. This approach allows the surveillance system to distinguish between normal and anomalous speeds.

The rest of this paper is organised as follows: Section 2 discusses several approaches and previous work related to speed analysis from visual information. Section 3 summarises the architecture of the surveillance system in which the speed analysis is carried out and a formal model to define new analysis components. In Section 4, we describe a novel method to classify the speed of moving objects as normal or abnormal, and how to integrate it into the global system. The experimental results are presented in Section 5. Finally, this paper is concluded in Section 6.

2. Previous work

As mentioned before, there are three relevant approaches to estimate the speed of moving objects from video information: (i) analysis of speed from aerial pictures, (ii) generation of a 3D representation of the monitored environment and, finally, (iii) use of a single camera in calibration processes and geometric methods to establish correspondences between 2D images and a portion of the real world.

Possibly, the first one is the simplest of these three approaches since it is not necessary to deal with the perspective problem and the number of occlusions is reduced. In this case, the movements of moving objects detected in a 2D image can be easily matched with the actual movements in the real world. Within this context, Liu et al. (2007) described a method to estimate the speed of vehicles on highways. To do that, the authors manually built a scene model in which it is possible to distinguish the highways from the rest of elements. Secondly, they make use of two consecutive aerial frames to establish correspondences between objects and calculate their speeds on the basis of their movements. Although this approach is an effective way to determine the speed, using aerial cameras it is not always possible, so other methods are required.

On the other hand, when the cameras do not provide aerial views, it is necessary to address the perspective problem, i.e., there is no direct equivalence between movements made in the

2D image and the actual ones. The approaches (ii) and (iii) deal with this problem. An expensive solution, but very accurate, is to build a 3D environment representation by using several 2D images obtained from different cameras. Cheung et al. (2000) proposed a system composed of five cameras which is able to perform 3D reconstruction of moving humans in real time. The system individually extracts the silhouettes of the moving people in each image and uses this information for generating a 3D reconstruction. Lee et al. (2000) proposed a method for monitoring activities from multiple video streams. Once the 3D model is built, the authors compare the speed and distance between objects in different parts of the scene. Hu et al. (2008) developed a system for traffic monitoring by data fusion from multiple stationary cameras. A probability fusion map is proposed to estimate the speed of vehicles.

Finally, many single-camera methods have been proposed as an alternative to the previously mentioned techniques. Cathey and Dailey (2005) presented a method for estimating the speed of vehicles on highways in Seattle. The cameras can be remotely controlled and every time their parameters are modified, a fast calibration process is performed. This process automatically detects the boundary lines of the road, which are used to determine the perspective of the scene. The proposed method is able to reduce the problem of perspective since it generates aerial views of the monitored scene by means of geometric methods. Once these images have been generated, the displacements of the vehicles are analysed to estimate their speeds in a third phase. Other similar methods in which the boundary lines of a highway are automatically detected, are discussed in Beymer et al. (1997), Palaio et al. (2009). Maduro et al. described in Maduro et al. (2008) how the previously cited methods lose their effectiveness when there are occlusions between objects, long distances covered by vehicles in few frames or low image resolution, among other reasons. They made a new proposal based on two previous methods (Cho & Rice, 2006; Magee, 2004). This new method can estimate the speed of a vehicle although it was not correctly detected by the tracking process.

The above methods have in common the calculation of a precise speed. From this calculated value, it is possible to know if an object moves between the allowed limits. A person watching a video is able to quickly determine whether a particular object is going too slow or fast depending on the area where it is located. However, this person would not be able to determine a precise value of the speed. In other words, a person usually refers to the speed and location of an object in terms of linguistic labels such as very left, very right, center, up, very down, etc. in the case of location, and very slow, slow, medium speed, fast or very fast, in the case of speed. This is the way in which we want our expert system to behave.

In the next section, we will briefly summarise the architecture of OCULUS surveillance system and the model on which it is based. OCULUS is a scalable and flexible surveillance system consisting of several independent normality components, that increases its analysis capability when new components are designed and included (Albusac, Vallejo, Jimenez-Linares, Castro-Schez, & Rodriguez-Benitez, 2009). A normality component specifies how each kind of object must behave according to a surveillance aspect such as trajectory or speed. The system includes a normality component for each surveillance aspect to be monitored and they can be enabled or disabled depending on the surveillance requirements. Besides, it combines the output of each component by means of aggregation operators in order to obtain a global interpretation of the current state of the environment. After describing the OCULUS architecture and the model on which it is based, the normality component designed to analyse the speed of objects will be presented in Section 4.

3. A scalable and flexible model

In previous work, we have addressed the problem of analysis behaviour and anomaly detection in urban environments (Vallejo, Albusac, Jimenez-Linares, Gonzalez-Morcillo, & Moreno, 2009). Specifically, the scheme presented in Vallejo et al. (2009) was designed to solve a specific problem in a particular environment: the behaviour analysis of vehicles and people on pedestrian crossings. Therefore, there is a lack of scalability to include new kinds of analysis, and a lack of flexibility to adapt the system to new environments. In order to overcome this problem, we designed a new architecture based on reusable components and a formal model to define them (Albusac et al., 2009). Fig. 1 shows an overview of this architecture, which is structured into three levels: (i) sensory level, (ii) reasoning level, and (iii) user level.

The sensory level contains the set of sensors distributed throughout the monitored environment. The information provided by these sensors is used by preprocessing modules whose main objective is to adapt the information to the normality components. For instance, the tracking preprocessor uses video information to detect and track moving objects. The information generated by this preprocessor is used by some normality components to carry out the behaviour analysis according to a surveillance aspect such as trajectory or speed.

The reasoning level consists of a set of normality components that analyse the behaviour of each kind of object. These components are defined in a general way and they can be instantiated for particular environments. For instance, the trajectory normality component is generally defined by specifying how a normal trajectory must be followed and what kind of constraints must be satisfied. Afterwards, the knowledge acquisition tools (KAT) and machine learning algorithms (MLA) associated to each component, make possible its particularisation for specific environments. Specifically, the particularisation of the trajectory component consists in defining the particular normal trajectories for the environment to be monitored.

Finally, the output of each normality component is combined to obtain an overall verdict about the current state of the monitored environment. When a new component is included into the system, the normal behavior of an object depends on a greater number of factors. For example, if only the trajectory component is activated, an object behaves normally if it follows a normal path. However, if the speed component is also activated, an object has not only to follow a normal path, but also to move at a normal speed.

The top level includes monitoring tools and visual alarms to draw the attention of the security staff when necessary. The monitoring tools also allow the security staff to watch an environment and to check the reasoning level response. Thus, it is possible to detect errors at the middle layer and solve them in order to improve the accuracy and robustness of the system. Next, we summarise a formal model to define normality components.

Although a previous version of this model has been described in detail in Albusac et al. (2009), it is our intention to make this paper reasonably self-contained. The version of the model described in this paper includes some modifications in order to improve it.

Definition 1. The surveillance of an environment E is the understanding of the different perceptions obtained from the sensors deployed on the monitored environment $\{S_1, S_2, \dots, S_n\}$. Each perception of the global environment is considered, at the same time, as an environment E_i . Thus, the global normality definition in the environment E is composed of the particular normality definitions for each sub-environment E_i . Therefore, the surveillance problem P in an environment E is defined as the monitoring of multiple sub-environments, which are simpler and developed from the perceptions of the sensors deployed on the environment:

$$P = \{E_1, E_2, \dots, E_n\} \tag{1}$$

In this way, the complexity of defining the normality is reduced since the number of situations that can take place in a sub-environment E_i is lower than in the global environment E .

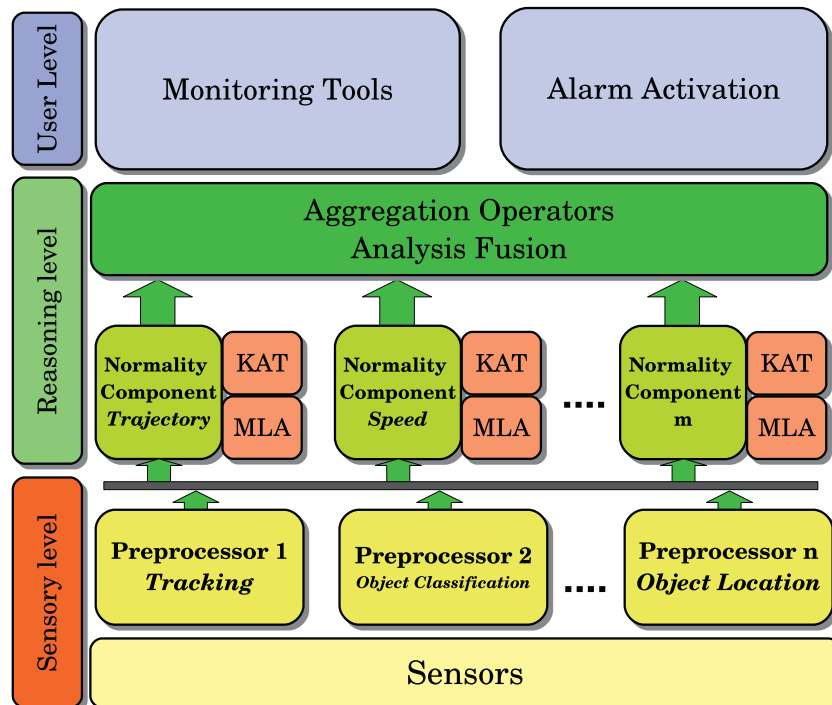


Fig. 1. OCULUS architecture.

Definition 2. A monitored sub-environment E is defined as a 4-tuple composed of the following elements:

$$E = \langle V; O; C; O \times C \rangle \quad (2)$$

where

- V is the set of input variables used to perform the required surveillance tasks and to inform about the objects features and the current state of the environment.
- O is the set of classes of monitored objects in the sub-environment, whose behaviour must be analysed (e.g. people, group of people, car, truck, bicycle, etc.).
- C refers to the set of monitored aspects in the sub-environment, denoted as *concepts* from now on.
- $O \times C$ determines the concepts that must be used to analyse the normality of each class of object. Depending on the class of each object, different concepts will be used to determine if their behaviour is normal or not.

Definition 3. A concept c_i ($c_i \in C$) is defined as a 3-tuple composed of the following elements:

$$c_i = \langle V_i; DDV_i; \Phi_i \rangle \quad (3)$$

where V_i is the set of input variables used to define the concept c_i so that $V_i \subseteq V$. On the other hand, DDV_i is the set of definition domains of the variables that belong to V_i . Therefore, if $V_i = \{v_{1i}, v_{2i}, \dots, v_{ni}\}$, then DDV_i is defined as $DDV_i = \{DDV_{1i}, DDV_{2i}, \dots, DDV_{ni}\}$, where DDV_{ji} is the definition domain of the variable v_{ji} . The definition domain of a variable specifies the possible values that can take. Finally, Φ_i is the set of constraints used to complete the definition of the concept c_i , according to the elements of V_i ($\Phi_i = \{\mu_{1i}, \mu_{2i}, \dots, \mu_{ki}\}$). The normality analysis of c_i depends on how the constraints associated to c_i are met.

Definition 4. A normality constraint, associated to a concept c_i is defined as a fuzzy set X_i over the domain $\mathcal{P}(V_i)$, with an associated membership function μ_{X_i} :

$$\mu_{X_i} : \mathcal{P}(V_i) \rightarrow [0, 1] \quad (4)$$

where 1 represents the maximum degree of satisfaction of the constraint and 0 the minimum. The rest of values represent intermediate degrees of normality.

The next step after defining a concept and its constraint in a general way is to make instances of such a concept for particular environments.

Definition 5. An instance y of a concept c_i in an environment E_j ($E_j \in P$), denoted as c_{iy}^j , is defined as follows:

$$c_{iy}^j = \langle V_i; DDV_i; \tilde{\Phi}_i = \{\tilde{\mu}_{1i}, \tilde{\mu}_{2i}, \dots, \tilde{\mu}_{zi}\} \rangle \quad (5)$$

where $\tilde{\Phi}_i$ is the set of particularised constraints of the set Φ_i , that is, each $\tilde{\mu}_{xi} \in \tilde{\Phi}_i$ represents the particularisation of $\mu_{xi} \in \Phi_i$. It is verified that $|\Phi_i| \geq |\tilde{\Phi}_i|$.

A normality constraint instance is used to adapt the general definition of a kind of analysis based on a concept to a specific environment.

Definition 6. A normality constraint instance is a fuzzy set defined over $\mathcal{P}(DDV_i)$ with an associated membership function ($\tilde{\mu}_{X_i}$):

$$\tilde{\mu}_{X_i} : \mathcal{P}(DDV_i) \rightarrow [0, 1] \quad (6)$$

so that if $v_{ki} \in \mathcal{P}(V_i)$ is employed to define μ_{X_i} , then the values of v_{ki} defined over $DDV_{ki} \in \mathcal{P}(DDV_i)$ are used to make the instance $\tilde{\mu}_{X_i}$.

Each monitored object has associated a degree of normality that establishes how normal its behaviour is according to each attached concept to it, represented by the deployed instances of such concepts within the environment.

Definition 7. The degree of normality of an object obj within an environment E_j ($E_j \in P$) regarding an instance y of the concept c_i (c_{iy}^j), denoted as $N_{c_{iy}^j}^j(obj)$, is calculated from the values obtained for each $\tilde{\mu}_{xi}$:

$$N_{c_{iy}^j}^j(obj) = \bigwedge_{x=1}^{|\Phi_i|} \tilde{\mu}_{xi}^y \quad (7)$$

being \bigwedge a t-norm, such as the t-norm that calculates the minimum value.

The next step after calculating the degree of normality of an object for each instance of a particular concept is to calculate the normality of such an object according to all the defined instances of such a concept. In this way, it is possible to study the general behaviour of an object according to a concept.

Definition 8. The degree of normality of an object obj within an environment E_j ($E_j \in P$) according to a concept c_i , denoted as $N_{c_i}^j(obj)$, is calculated as follows:

$$N_{c_i}^j(obj) = \bigvee_{y=1}^w N_{c_{iy}^j}^j(obj) \quad (8)$$

where w is the number of instances of the concept c_i and \bigvee is a t-conorm operator, for instance the maximum t-conorm.

On the other hand, considering the application of the t-norm and t-conorm operators, the normality analysis of an object obj , according to c_i in E_j , is the result of applying an AND-OR fuzzy network over a set of constraints:

$$\begin{array}{ccccccc} c_{i1}^j & = & \tilde{\mu}_{1i}^1 \wedge \tilde{\mu}_{2i}^1 \wedge \dots \wedge \tilde{\mu}_{ni}^1 & = & N_{c_{i1}^j}^j(obj) \\ & & & & \downarrow \\ c_{i2}^j & = & \tilde{\mu}_{1i}^2 \wedge \tilde{\mu}_{2i}^2 \wedge \dots \wedge \tilde{\mu}_{ni}^2 & = & N_{c_{i2}^j}^j(obj) \\ & & \cdot & & \cdot \\ & & \cdot & & \cdot \\ & & \cdot & & \cdot \\ & & \cdot & & \cdot \\ c_{iw}^j & = & \tilde{\mu}_{1i}^w \wedge \tilde{\mu}_{2i}^w \wedge \dots \wedge \tilde{\mu}_{ni}^w & = & N_{c_{iw}^j}^j(obj) \\ \hline & & & & N_{c_i}^j(obj) \end{array}$$

In short, the degree of normality of an object associated to a concept c_i , $N_{c_i}^j(obj)$ is a numerical value that belongs to the interval $[0, 1]$, which is a representative sign of the object behaviour regarding a concept or surveillance aspect. High values of this parameter represent normal situations within the monitored environment while low ones represent suspicious or abnormal situations.

The final goal to reach after the analysis of a situation is to activate a set of alarms or to draw the attention of the security staff when such a situation does not meet the limits of normality. That is, after having calculated the degree of normality $N_{c_i}^j(obj)$, the model needs a mechanism to decide whether the object behaves normally or not, depending on the calculated degree of normality. The normality is considered as a linguistic variable V_N that takes a set of values over the domain definition $DDV_{V_N} = \{AA, PA, SB, PN, AN\}$ (see Fig. 2). In this way, the object behaviour can be absolutely abnormal (AA), possibly abnormal (PA), suspicious (SB), possibly normal (PN), and absolutely normal (AN); so that a behaviour can belong to more than one set at the same time.

The definition of each value of the domain DDV_{V_N} depends on the features of the environment to monitor, the desired security level and the criterion of the expert in charge of setting up the con-

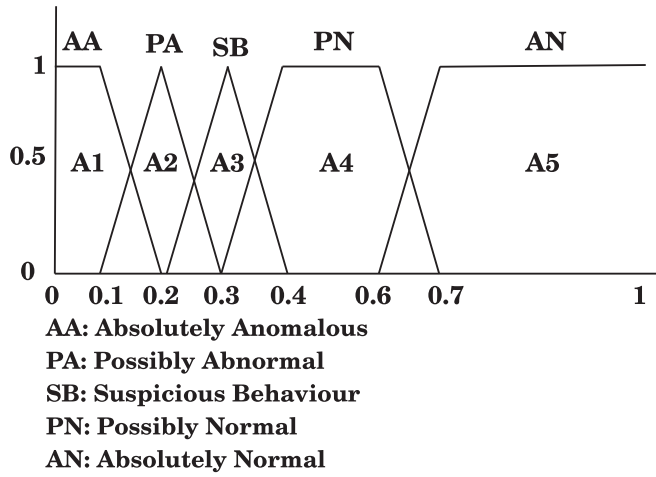


Fig. 2. Definition domain DDV_{V_N} of the normality variable V_N .

figuration parameters, which must be adapted to the system behaviour. Fig. 2 graphically shows how the values assigned to the sets that represent anomalous situations are not high, which implies that the system is not very strict, avoiding in this way a frequent (and possible unnecessary) alarm activation.

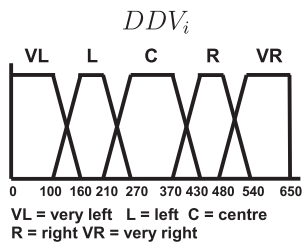
In this way, every time that the degree of normality of an object behaviour is calculated according to a concept, $N^j_{c_i}(obj)$, the membership of this value to the fuzzy sets that establish the definition domain DDV_{V_N} of the normality variable (V_N) is studied, determining the normality of the analysed situation. The alarm activation will rely on upper layers, which will perform the required actions depending on the normality values estimated.

On the other hand, the normality of an object does not exclusively depend on a concept. In fact, a global evaluation of the

Table 1

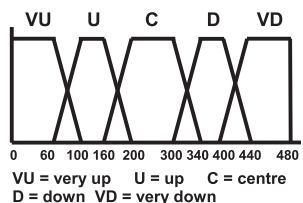
Set of variables used to determine the horizontal and vertical location of an object. The values established for each fuzzy set can easily vary depending on the scene to be monitored and the image resolution.

Variable V_i	DDV_i	Description
HP		Horizontal position



VP

Vertical position



normality of all the monitored concepts within the environment must be carried out. Therefore, a mechanism for combining multiple analysis is needed to get a global value that represents the normality of the object behaviour in a general way. The use of OWA (ordered weighted averaging) operators (Yager, 1988) is proposed to address this problem due to their flexibility.

Formally, an OWA operator is represented as a function $F: R^n \rightarrow R$ associated to a vector of weights W of length n ; $W = [w_1, w_2, \dots, w_n]$, where each $w_i \in [0, 1]$ and $\sum_{i=1}^n w_i = 1$.

$$OWA(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j \cdot b_j \tag{9}$$

$$OWA(a_1, a_2, \dots, a_n) = [w_1, w_2, \dots, w_n] \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

where (a_1, a_2, \dots, a_n) represent the set of initial values or criteria used by the operator to make a decision, and (b_1, b_2, \dots, b_n) represent the ordered set associated to (a_1, a_2, \dots, a_n) , being b_j the j th highest value of such a set. Furthermore, the values of the weights w_i belonging to the vector W are linked to positions and not to particular values of the original set.

In the model devised in this work, the OWA operator is used to aggregate the normality values calculated by each component $(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n})$. One of the key characteristics of this set of aggregation operators is the flexibility to vary their behaviour depending on the values assigned to the vector of weighs W . Such a vector determines the behaviour of the operator, which may tend to behave as union or intersection operators. In fact, this behaviour can be customised to reflect the minimum t-norm or the maximum t-conorm.

After having obtained the global normality value by applying the OWA operator $(OWA(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n}))$, the final normality value $N^j(obj_k)$ associated to an object obj_k in a particular environment E is given by the degree of membership of this value to the fuzzy sets mentioned before. These sets define the values *possibly normal behaviour* (PN) and *absolutely normal behaviour* (AN) of the definition domain DDV_{V_N} of the normality variable V_N (see Fig. 2), determining in this way the normality of the analysed situation:

$$N^j(obj_k) = \mu_{PN}(OWA(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n})) + \mu_{AN}(OWA(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n})) \tag{10}$$

where $\mu_{PN}(OWA(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n}))$ and $\mu_{AN}(OWA(N^j_{c_1}, N^j_{c_2}, \dots, N^j_{c_n}))$ establish the membership of the output value of the OWA operator to the sets *PN* and *AN*, respectively.

The global normality value within an environment E_j considering the activity of all the moving objects, denoted as GN^j , is calculated as the minimum of the normality value N^j calculated for each object obj_k :

$$GN^j = \bigwedge_{k=1}^{|O|} N^j(obj_k) \tag{11}$$

where O is the set of monitored objects at a particular time, $|O|$ is the number of monitored objects and $N^j(obj_k)$ is the normality value calculated for each object $obj_k \in O$.

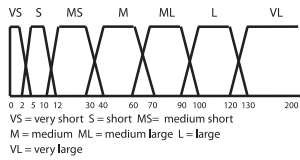
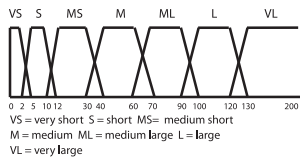
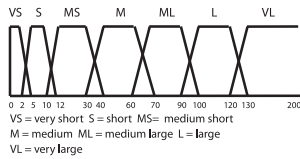
The use of the minimum is justified because when an object does not behave normally, the global normality degree GN^j in the environment E_j must be a low value.

Table 2
Variables used to represent the membership of an object to different classes as well as the list of allowed classes of objects that move at a particular speed. Each environment has a set of instances of the speed concept, and each can only be satisfied by a set of object classes (Υ).

Variable (V_i)	DDV_i	Description
$\mu_c(obj)$	[0, 1]	Fuzzy information about the membership of an object to one class c , $\forall c \in O, \exists \mu_c(obj)$
Υ	O	Υ represents the set of classes/roles that are allowed to move at a particular speed. Each speed instance has an associated Υ , which it is used to define the role constraints
v_c	$[0, (O - 1)]$	The value of this variable represents the identifier of the class to which the object belongs. If the set of object classes O consists of four classes then the domain of v_c would be $DDV_{v_c} = \{0, 1, 2, 3\}$

Table 3
Set of variables to determine the displacement of an object. As shown in Table 1, the numeric values that represent the limits of the fuzzy set can be modified depending on the scene to be monitored and the criterion of the expert who is responsible for maintaining the knowledge base. In this work, the displacements are analysed each 25 frame due to the scene characteristics. However in other scenes might be necessary to perform the analysis in a shorter time interval.

Variable V_i	DDV_i	Description
HD		Horizontal displacement
VD		Vertical displacement
MOV		Global displacement



- What objects are in the scene. Objects that appear in the monitored scene.
- Location of each detected object (Table 1).
- Class to which each object belongs. The class of an object is an important factor to determine if its speed is normal or not. For example, a normal speed for a vehicle may represent an excessive speed for a person (Table 2).
- Horizontal, vertical and global displacements in the 2D image (Table 3).
- Normal displacements allowed in each region of the 2D image for each object class.

Fig. 3 shows the relationship between the preprocessors and the speed normality component. In the following sections, we explain how the normality component represents and makes use of the input information to perform its main function of classifying and labelling the speed of an object.

4.1. Definition of variables (V_i)

The set of variables V_2 that defines the speed concept¹ c_2 is organised into four groups: (a) variables that determine the horizontal and vertical locations of an object, (b) variables that allow to establish the membership of an object to each class, (c) variables that provides information about the displacements in the 2D image, and finally (d) variables exclusively used to define spatial constraints.

(a) Horizontal and vertical location of an object in the 2D image.

First, a horizontal and vertical division² of the 2D image is needed to be made as shown in Fig. 4. Each sector has associated a linguistic label such that an object may be horizontally located at (VL) very left, (L) left, (C) centre, (R) right, (VR) very right, or vertically located at (VU) very up, (U) up, (C) centre, (VD) very down, or (D) down.

To calculate the horizontal and vertical position of an object, the system studies the membership of its central point³ (x, y) to the fuzzy sets defined in the division of the 2D image. Table 1 shows the variables that represent the horizontal and vertical position of an object as well as their domains. It is important to remark that the values shown in Table 1 for the fuzzy sets correspond to the scene chosen for testing. However, these values may vary depending on the scene and the criterion of the expert who has to make the image division.

4. Speed normality component

The model described in the previous section is employed to define new normality components. The design of a new component implies the correct choice of a set of variables and constraints that allow to study the aspect in which the component is based. The value of the variables that are part of the constraints can be directly provided by the distributed sensors (e.g. video cameras) or by the preprocessors (e.g. tracker) located in the lower layers. Specifically, the component described in this section needs the following information to classify the speed as normal or abnormal, which is provided by preprocessors located in the lower layers:

¹ There is a component c_1 included in OCULUS system to analyse trajectories. For this reason, this new component is called as c_2 .

² Currently, this division is made by hand, however an algorithm to automatically determine the best division of an image will be presented in future work. This algorithm takes into account the size of objects and areas where they move more frequently.

³ The tracking process integrated into the OCULUS system envelopes each detected moving object by means of an ellipse. The central point of this ellipse is used to estimate the object location.

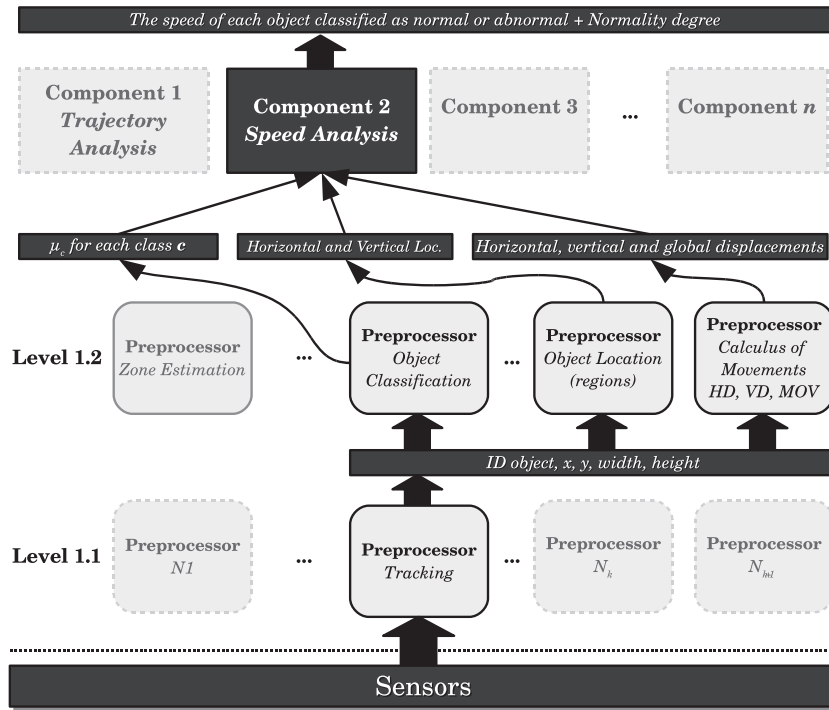


Fig. 3. This scheme shows how the processors are connected to each other. The output of the preprocessors located at level 1.2 is input for the speed normality component. This component knows the detected objects in the scene at each key moment, where they are located and the magnitude of their movements.

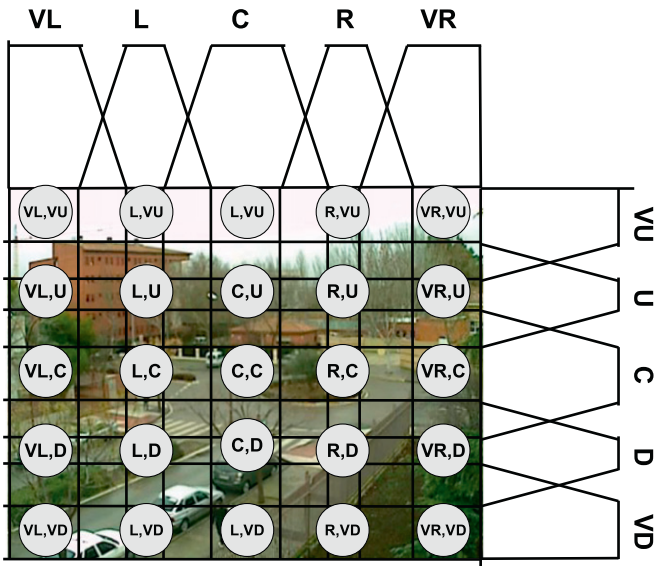


Fig. 4. Example of a vertical and horizontal division of a 2D image which represents a scene to be monitored. The meaning of the linguistic labels used to represent the horizontal position is as follows: VL: very left, L: left, C:center, R: right, VR: very right. On the other hand, the meaning of the linguistic labels used to represent the vertical position is as follows: VU: very up, U: up, C: centre, D: down, VD: very down. Each circle represents an area of the 2D image in which the membership to the fuzzy sets is maximum.

(b) *Class to which each object belongs.* Each $\mu_c(obj)$ represents the membership of an object obj to a class c , and it belongs to the interval $[0, 1]$. The algorithm to classify moving objects as people or vehicles has been explained in Albusac, Castro-Schez, Lopez-Lopez, Vallejo, and Jimenez-Linares (2009). This algorithm classifies moving objects from a set of fuzzy rules which have been automatically generated by a supervised learning algorithm. The domain of the variable $\mu_c(obj)$

can be seen in Table 2. This table also includes the description of the variable Υ which is used to define role constraints. This variable is associated to each speed instance and determines the set of objects that can move at a particular speed. Finally, v_c contains the identifier of the class whose $\mu_c(obj)$ is maximum. When an object is classified, it can have a membership value greater than zero in different classes. An object is classified into the class whose $\mu_c(obj)$ is maximum.

(c) *Object displacements.* Whenever an object moves in the scene, the central point of this object varies its position in the vertical and horizontal axis of the 2D image. If (x_1, y_1) represent the coordinates of an object obj at a time instant t_1 , and (x_2, y_2) the coordinates of the same object at t_2 , being $t_2 > t_1$, the displacements made by obj in the 2D image are calculated as follows:

- Horizontal displacement: $HD = \frac{\Delta x}{\Delta t} = \frac{|x_2 - x_1|}{t_2 - t_1}$.
- Vertical displacement: $VD = \frac{\Delta y}{\Delta t} = \frac{|y_2 - y_1|}{t_2 - t_1}$.
- Global displacement: $MOV = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{\Delta t} = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{t_2 - t_1}$.

After having calculated the numeric values for the above variables, the speed component fuzzifies them in order to obtain a linguistic representation. These variables and their domains are summarised in Table 3.

(d) *Variables used to defined spatial constraints.* Table 4 shows the rest of variables used to define spatial constraints. Each of these variables represents a subset of the domains of the previous variables (DDV_i) and determine the possible horizontal and vertical position as well as the vertical, horizontal and global displacements allowed for a particular instance c_{iv}^j of the concept *normal speed* c_2 .

4.2. Definition of constraints (Φ_i)

Once the set of variables V_i has been defined, the second step is to define the set of constraints Φ_i from the combination of these

variables. These constraints are initially defined in a general way, and they are later instantiated for particular environments. Each instance represents a normal speed in the monitored environment. If an object satisfies the constraints of at least one instance, then it moves at a normal speed.

Role constraint (μ_{12}). Each instance c_{ij}^i of the speed concept c_2 has associated a list Υ that contains the object classes for which the speed is appropriate. An object satisfies the role constraint when it is classified as a class of Υ . The membership function for this constraint is defined as follows:

$$\mu_{12}(obj) = \begin{cases} 1 & \text{if } v_c \text{ is an identifier of a class } c \in \Upsilon \\ 0 & \text{otherwise.} \end{cases}$$

Such that v_c represents the identifier of the class c whose $\mu_c(obj)$ is maximum. The remaining constraints are spatial constraints defined as follows:

Table 4

List of the rest of variables used to define spatial constraints for the speed normality component.

Variable (V_i)	DDV_i	Description
$\ell_{HP}(obj)$	DDV_{HP}	Set of linguistic labels belonging to DDV_{HP} that determine the horizontal positions where an object has to be located to move at a particular speed
$\ell_{VP}(obj)$	DDV_{VP}	Set of linguistic labels belonging to DDV_{VP} that represent the vertical positions where an object has to be located to move at a particular speed
$\ell_{HD}(obj)$	DDV_{HD}	Allowed horizontal movements for an instance of the concept <i>normal speed</i>
$\ell_{VD}(obj)$	DDV_{VD}	Allowed vertical movements for an instance of the concept <i>normal speed</i>
$\ell_{MOV}(obj)$	DDV_{MOV}	Global movements for an instance of the concept <i>normal speed</i>

Horizontal location of an object (μ_{22}). This kind of constraint checks whether an object is located at a particular horizontal region of the 2D image. The component fuzzifies the numeric value of the variable HP in order to obtain one or more linguistic labels that represent the current horizontal position of the object. An object satisfies this constraint just in case of these labels belong to the regions defined in the list ℓ_{HP} for an instance of the concept *normal speed*. The list ℓ_{HP} is actually a subset of the set {VL: very left, L: left, C: centre, R: right, VR: very right}, i.e., $\ell_{HP} \subseteq DDV_{HP}$ (see Table 1). The degree of satisfaction of this constraint is calculated as the sum of all membership values to the sets included in ℓ_{HP} :

$$\mu_{22}(obj) = \sum_{hz \in \ell_{HP}} \mu_{hz}(obj)$$

where $\mu_{Hz}(obj)$ represents the membership value of the object obj to the horizontal region H_z , being H_z a region included in ℓ_{HP} . Fig. 5 shows how to estimate the degree of satisfaction of this kind of constraints.

Vertical location of an object (μ_{32}). This constraint checks whether an object is located at a particular vertical region of the 2D image. The definition of this constraint is similar to μ_{22} , but the list of regions is now $\ell_{VP} \subseteq DDV_{VP}$. Therefore, μ_{31} is defined as follows:

$$\mu_{32}(obj) = \sum_{vz \in \ell_{VP}} \mu_{vz}(obj)$$

where $\mu_{hd}(obj)$ represents the membership of the object obj to the vertical region vz , being vz a region included in ℓ_{VP} . The degree of satisfaction of the constraint $\mu_{32}(obj)$ is calculated as the sum of all $\mu_{vz}(obj)$ such that $vz \in \ell_{VP}$.

Horizontal movement of an object (μ_{42}). This constraint checks the magnitude of the horizontal movement made by an object. The variable ℓ_{HD} represents the set of allowed horizontal displacements for a particular instance of the concept *normal speed*. The linguistic labels of this set belong to the domain $DDV_{HS}\{V_S$:

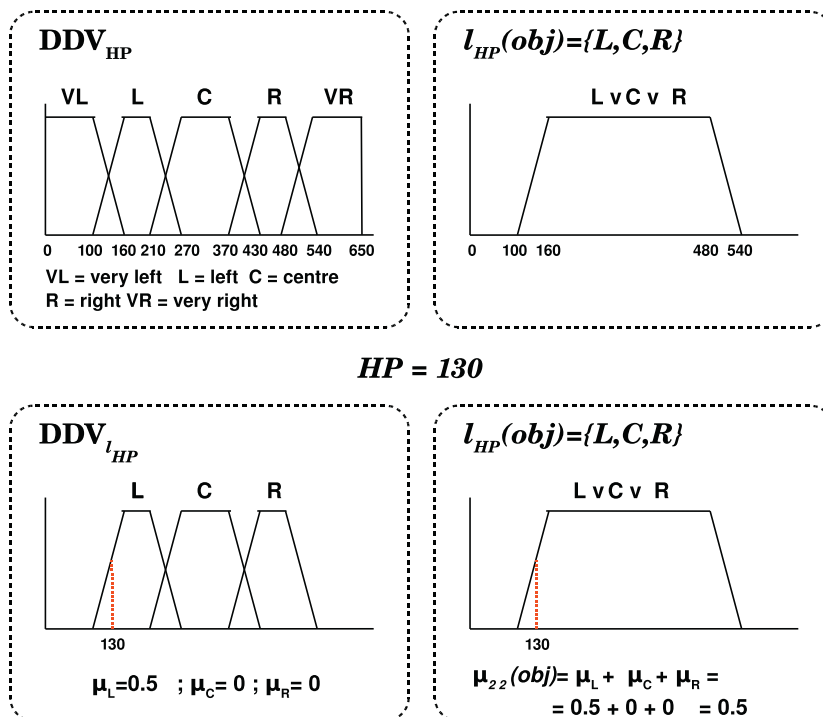


Fig. 5. Calculation of degree of satisfaction for the horizontal location constraint, being $HP = 130$ and $\ell_{HP} = \{L, C, R\}$. In this case, the satisfaction degree of the constraint $\mu_{22}(obj)$ is 0.5.

very short, S: short, MS: medium short, M: medium, L: medium large, L: large, VL: very large}, i.e., $\ell_{HD} \subseteq DDV_{HD}$. The variable HD contains the value of the last horizontal displacement made by an object along the axis X . The component c_2 studies the membership of the variable HD to the fuzzy sets defined in ℓ_{HD} , and the satisfaction degree is calculated as the sum of all degrees of membership:

$$\mu_{42}(obj) = \sum_{hd \in \ell_{HD}} \mu_{hd}(obj)$$

such that $\mu_{hd}(obj)$ represents the membership value of the object obj to the displacement hd included in the set ℓ_{HD} .

Vertical movement of an object (μ_{52}). This constraint checks the magnitude of the vertical movement made by an object. The variable ℓ_{VD} represents the set of allowed vertical displacement for a particular instance of the concept *normal speed*. As in HD , the variable VD contains the value of the last vertical displacement made by an object along the axis Y . The component c_2 also studies the membership of VD to the fuzzy sets defined in ℓ_{VD} , and the degree of satisfaction is also calculated as the sum of all degrees of membership:

$$\mu_{52}(obj) = \sum_{vd \in \ell_{VD}} \mu_{vd}(obj)$$

such that $\mu_{vd}(obj)$ represents the membership value of the object obj to the displacement vd included in the set ℓ_{VD} .

Global movement of an object (μ_{62}). This constraint checks the magnitude of the global displacement made by an object between consecutive time instants, taking into account both kind of movements: horizontal and vertical. The variable MOV contains the value of the last global movement calculated as described in Section 4.1. To check the magnitude of the global movement, the component c_2 studies the membership of the value MOV to the fuzzy sets defined in ℓ_{MOV} . The degree of satisfaction of μ_{62} is calculated as follows:

$$\mu_{62}(obj) = \sum_{mov \in \ell_{MOV}} \mu_{mov}(obj)$$

being $\mu_{mov}(obj)$ the membership of MOV to the fuzzy set mov included in ℓ_{MOV} .

Once the constraints have been defined in a general way, the normality component can be particularised for specific environments. The next Section describes how the speed component can be instantiated and some instances of c_2 are presented as examples.

4.3. Particularisation of the concept normal speed

The speed component, as any other normality component, needs specific knowledge about the environment to be analysed. This allows the adaptation of the general definition of the concept c_2 to a particular environment. In this case, the component c_2 needs to know the allowed displacements between consecutive time instants, in each region of the 2D image for each kind of object. The definition of multiple instances of the speed concept for a concrete camera is a tedious, time-consuming and error-prone task. Thus, these instances should be learned automatically. For this reason, we propose a supervised learning algorithm to speed up the deployment of new cameras in an environment. Really, this algorithm is a modification of Castro, Castro-Schez, and Zurita (1999), which has been properly adapted to the problem of surveillance and described in Albusac et al. (2009).

Basically, the algorithm consists of three main stages: (a) an expert defines the domain of each input variable through several regions and trapezoidal functions, (b) each sample of the training set becomes an initial rule, and (c) a reduced set of general rules is

generated from the initial set of rules. The fuzzy IF-THEN rules generated by the algorithm are as follows:

$$\text{if } v_0 \text{ is } ZD_0 \wedge \dots \wedge v_n \text{ is } ZD_n \text{ then } y_j \quad (12)$$

where $v_i \in V$ and $ZD_i \subseteq DDV_i$ is a subset of the linguistic labels defined in DDV_i for the variable v_i . Next, we formally describe the algorithm in Algorithm 1.

Algorithm 1. Machine learning algorithm.

Require Training set Θ , input variables, and a domain definition for each input variable.

Ensure Set of definitive rules.

- (1) Convert each training example $e_i = ((x_{i1}, \dots, x_{in}), y_j)$ into an initial fuzzy rule (translation into the fuzzy domain). Each element $e_i \in \Theta$ is translated into a fuzzy rule in which the value of each input variable is represented by means of a linguistic label. This step creates the set of *initial* rules and an empty set of definitive rules.
 - (2) Take a rule from the set of initial rules. Let R_i be that rule, R_i : if v_0 is $ZD_{i0} \wedge \dots \wedge v_n$ is ZD_{in} then y_p .
 - (3) Try to subsume the taken rule R_i in some rule of the set of definitive rules. If that happens so, ignore the taken rule and go back to step 2 (the concept *subsume* is described in Castro et al. (1999)).
 - (4) If the taken rule does not subsume in any rule of the definitive set, try to amplify it. For each variable v_k :
 - (4.a) For each unconsidered label L_x :
 - (4.a.i) Try to amplify the rule $ZD_{i'k} = ZD_{ik} \cup L_x$. If it is not possible, go to step 4.a; otherwise, proceed to step 4.a.ii. One rule can be amplified only if:
 1. There is no R_j : **if** v_0 is $ZD_{j0} \wedge \dots \wedge v_n$ is ZD_{jn} **then** y_q in the set of initial rules, such that $\forall k \in [0, |V|]$

$$ZD_{jk} \subseteq ZD_{i'k} \text{ and } y_p \neq y_q$$
 2. The separability between the last label added to the rule and the label being considered for amplifying the rule does not exceed a separability threshold. If $(a_{\underline{L}_i}, b_{\underline{L}_i}, c_{\underline{L}_i}, d_{\underline{L}_i})$ and $(a_{\underline{L}_j}, b_{\underline{L}_j}, c_{\underline{L}_j}, d_{\underline{L}_j})$ represent the trapezoids that define the fuzzy sets L_i and L_j respectively, the separability between them is calculated as follows:

$$\text{separability}(L_i, L_j) = \frac{(b_{L_j} - c_{L_i}) + (a_{L_j} - d_{L_i})}{2} \quad (13)$$
 - (4.a.ii) Amplify the rule (the amplification concept is also described in Castro et al. (1999)).
 - (5) If the taken rule has been amplified, then include the rule in the set of definitive rules.
 - (6) If there are still unconsidered rules in the initial set of rules, go to step 2. Otherwise, **END**.
-

There is a direct correspondence between the rules generated by the learning algorithm and the instances of the *normal speed* concept. Each rule represents an instance of this concept. Table 5 shows an example of a normal speed which has been built from the particularisation of the variables and constraints defined in Sections 4.1 and 4.2, respectively.

An object satisfies the constraints of the instance shown in Table 5, if the object is a vehicle horizontally located at *center* or *right*, vertically located at *down* or *very down*, its last horizontal displacement has been *very short*, *short* or *medium short*, its last vertical displacement has been *very short* and, finally, its last global displacement has been *very short*, *short* or *medium short*. In other words, if an object satisfies these six constraints, then the system

Table 5

Example of a normal speed instance. This instance is created from the particularisation of the variables and constraints defined for the normality component.

$\tilde{\mu}_{12}$	$\tilde{\mu}_{22}$	$\tilde{\mu}_{32}$	$\tilde{\mu}_{42}$	$\tilde{\mu}_{52}$	$\tilde{\mu}_{62}$
$\Upsilon = \{\text{vehicle}\}$	$\ell_{HP} = \{C, R\}$	$\ell_{VP} = \{VD, D\}$	$\ell_{HD} = \{VS, S, MS\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{VS, S, MS\}$

Table 6

Samples from the training set used to generate the definitive rules. In this particular case, the samples are divided into two classes: normal and abnormal speed. The objects are classified as pedestrian or vehicle, whose identifiers are 0 and 1, respectively. Each sample is represented by means of numeric values for the following variables: $((x, y, \text{class identifier}, \text{HD}, \text{VD}, \text{MOV}), y_j)$. The class normal speed and abnormal speed are denoted as y_1 and y_2 , respectively.

Normal speed	Abnormal speed
$((388, 324, 0, 19.0, 4.0, 19.41), y_1)$	$((19, 371, 1, 131.0, 47.0, 139.17), y_2)$
$((386, 332, 0, 10.0, 0.0, 10.0), y_1)$	$((564, 292, 1, 168.0, 14.0, 168.58), y_2)$
$((186, 307, 1, 42.0, 3.0, 42.10), y_1)$	$((568, 261, 1, 126.0, 3.0, 126.03), y_2)$
$((114, 340, 1, 90.0, 31.0, 95.18), y_1)$	$((497, 283, 0, 90.0, 5.0, 90.13), y_2)$
$((200, 250, 1, 52.0, 12.0, 53.36), y_1)$	$((492, 253, 0, 75.0, 11.0, 75.80), y_2)$

Table 7

Labels for the samples in Table 6. Each represents an initial rule from which a definitive rule might be generated.

Normal speed	Abnormal speed
$((C, D, \text{Pedestrian}, MS, S, MS), y_1)$	$((VL, D, \text{Vehicle}, VL, M, VL), y_2)$
$((C, D, \text{Pedestrian}, S, VS, S), y_1)$	$((VR, C, \text{Vehicle}, VL, MS, VL), y_2)$
$((L, C, \text{Vehicle}, M, VS, M), y_1)$	$((VR, C, \text{Vehicle}, VL, VS, VL), y_2)$
$((VL, D, \text{Vehicle}, ML, MS, L), y_1)$	$((R, C, \text{Pedestrian}, ML, S, ML), y_2)$
$((L, C, \text{Vehicle}, M, MS, M), y_1)$	$((R, C, \text{Pedestrian}, ML, S, ML), y_2)$

can infer that such an object moves at the speed defined in Table 5 and, therefore, the object moves at a normal speed.

Each column in Table 5 represents an instantiation of a constraint in which the variables take particular values from their domains. The fuzzy rule equivalent to the instance shown in Table 5 is as follows:

IF $\Upsilon = \{\text{vehicle}\} \wedge \ell_{HP} \text{ is } \{C, R\} \wedge \ell_{VP} \text{ is } \{VD, D\} \wedge \ell_{HD} \text{ is } \{VS, S, MS\} \wedge \ell_{VD} \text{ is } \{VS\} \wedge \ell_{MOV} \text{ is } \{VS, S, MS\}$ **THEN** y_j is normal speed.

In this component, the instances of the concept c_2 can be easily represented by means of IF-THEN fuzzy rules, but this fact does not mean that different constraints included in other different normality components can be represented in the same way. The model de-

Table 8

Examples of fuzzy rules to classify the speed of objects as normal or abnormal.

<i>Normal speed</i>
R_1 : IF y is $\{VD, D\} \wedge vd$ is $\{VS, S\}$ THEN normal speed
R_2 : IF c is $\{\text{Vehicle}\} \wedge mov$ is not $\{L, VL\}$ THEN normal speed
R_3 : IF x is not $\{VR\} \wedge vd$ is $\{VS\}$ THEN normal speed
R_4 : IF x is $\{R, VR\} \wedge c$ is $\{\text{Vehicle}\} \wedge vd$ is $\{S\}$ THEN normal speed
R_5 : IF c is $\{\text{Vehicle}\} \wedge hd$ is not $\{L, VL\}$ THEN normal speed
<i>Abnormal speed</i>
R_6 : IF y is $\{D, VD\} \wedge hd$ is $\{L, VL\}$ THEN abnormal speed
R_7 : IF x is not $\{D, VD\} \wedge c$ is $\{\text{Pedestrian}\} \wedge mov$ is not $\{VS, S, SM\}$ THEN abnormal speed
R_8 : IF x is $\{VL, L\} \wedge y$ is not $\{D, VD\} \wedge mov$ is $\{L, VL\}$ THEN abnormal speed
R_9 : IF x is $\{C\} \wedge y$ is $\{C\} \wedge c$ is $\{\text{Vehicle}\} \wedge hd$ is $\{L\} \wedge vd$ is $\{S\} \wedge mov$ is $\{L\}$ THEN abnormal speed
R_{10} : IF x is $\{VL\} \wedge y$ is $\{D\} \wedge c$ is $\{\text{Human}\} \wedge hd$ is $\{LM\} \wedge vd$ is $\{SM\} \wedge mov$ is $\{LM\}$ THEN abnormal speed

scribed in Section 3 is based on defining the normality of an environment from a set of constraints and not from a simple system of rules. Some constraints may involve more complex definitions such as the spatial and temporal constraints defined in the trajectory component (Albusac et al., 2009).

Finally, Table 6 shows some samples from the training set used to generate the set of rules/instances. These samples are fuzzified and converted into initial rules such as Table 7 shows. These samples are lately amplified into general rules (some of these examples are shown in Table 8). Table 9 exposes the direct correspondence between the rules in Table 8 and the instances of the concept c_2 . All required steps to obtain the set of definitive rules are graphically illustrated in Fig. 6.

5. Results

The scenario chosen for validating and evaluating the model and the designed normality components is graphically shown in Fig. 7. It is an urban traffic environment where both vehicles and pedestrians are submitted to traffic laws, such as correct trajectories or suitable speed for each stretch. This environment is monitored by means of a security camera located in the ORETO research lab at the University of Castilla-La Mancha. The process of experimental validation consists in monitoring 10 video scenes whose duration ranges from 30 to 180 s. Figs. 8 and 9 show the degree of illumination of each one of the scenes employed to test the component. As can be seen, the method proposed in this work has been tested under different illumination conditions.

Each one of the detected objects in a single frame is considered as a situation to be analysed. For each one of these situations, the system checks if the classification, regions calculation, trajectory analysis, speed analysis, and global analysis are normal. The matches and errors of the normality components and the global analysis are classified into true positives (TP), true negatives (VN), false positives (FP), and false negatives (FN); each one of them representing the following situations:

- **True positive (TP)**: normal situation correctly classified as normal.
- **True negative (TN)**: anomalous situation correctly classified as anomalous.
- **False positive (FP)**: anomalous situation incorrectly classified as normal.
- **False negative (FN)**: normal situation incorrectly classified as anomalous.

Furthermore, for each one of the evaluated tests, the relationships between these parameters have been established by means of the following coefficients (Fawcett, 2003):

- Sensitivity or true positive rate (TPR): $TPR = TP/(TP + FN)$, that is, the hit rate.
- False negative rate (FNR): $FNR = FN/(FP + FN)$.
- Precision or accuracy (ACC): $ACC = (TP + TN)/(P + N)$, where P is the number of positive cases and N the number of negative cases.
- Specificity (SPC) or true negative rate: $SPC = TN/N = TN/(FP + TN) = 1 - FPR$.
- Positive predictive value (PPV): $PPV = TP/(TP + FP)$.
- Negative predictive value (NPV): $NPV = TN/(TN + FN)$.
- False discovery rate (FDR): $FDR = FP/(FP + TP)$.

Table 10 shows the results obtained in the speed classification process. In this case, the average success rate is 98.7%. In the con-

Table 9

Examples of the concept c_2 which have been learned by means of the learning algorithm previously described. These instances correspond with the normal speed rules shown in Table 8.

$c_{2,y}^1$	$\tilde{\mu}_{t12}$	$\tilde{\mu}_{t22}$	$\tilde{\mu}_{t32}$	$\tilde{\mu}_{t42}$	$\tilde{\mu}_{t52}$	$\tilde{\mu}_{t62}$
$c_{2,1}^1$	$\Upsilon = \{\emptyset\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{VD, D\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS, S\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,2}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{\emptyset\}$	$\ell_{MOV} = \{VS, S, MS, M, ML\}$
$c_{2,3}^1$	$\Upsilon = \{\emptyset\}$	$\ell_{HP} = \{VL, L, C, R\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,4}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{R, VR\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,5}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{VS, S, MS, M, ML\}$	$\ell_{VD} = \{\emptyset\}$	$\ell_{MOV} = \{\emptyset\}$

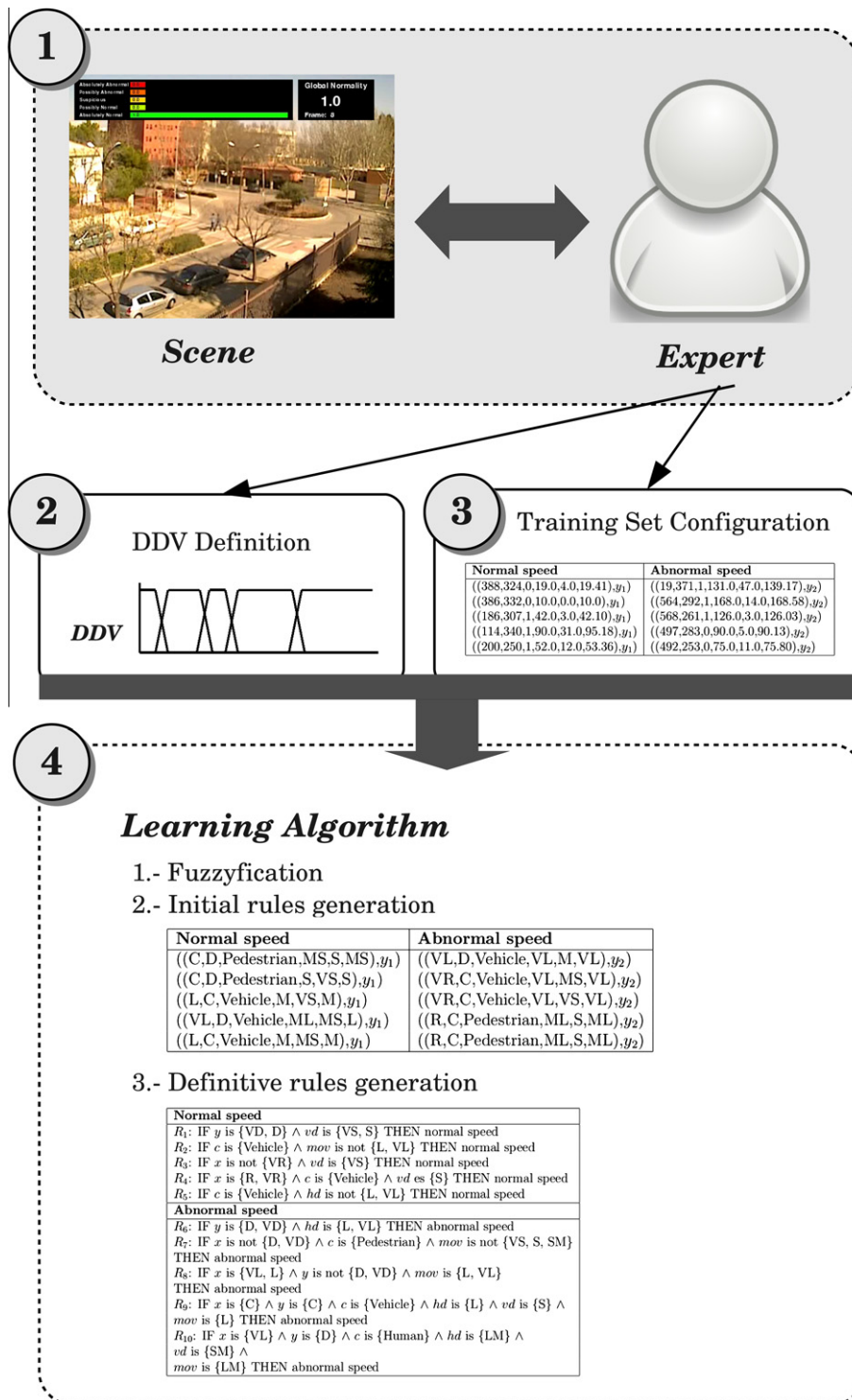


Fig. 6. Sequence of the steps required to obtain the set of definitive fuzzy rules used for classifying the speed from 2D images.

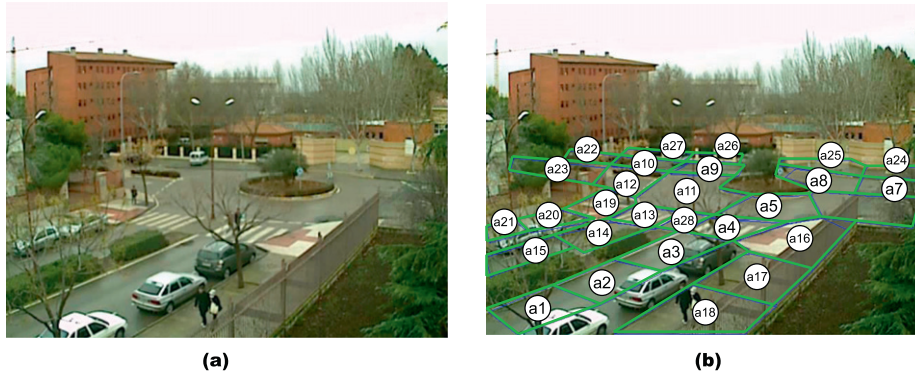


Fig. 7. (a) Scene monitored by means of a security camera. (b) Scene division into areas or regions needed for the trajectory normality component.



Fig. 8. Screenshots of the scenes employed in the tests.

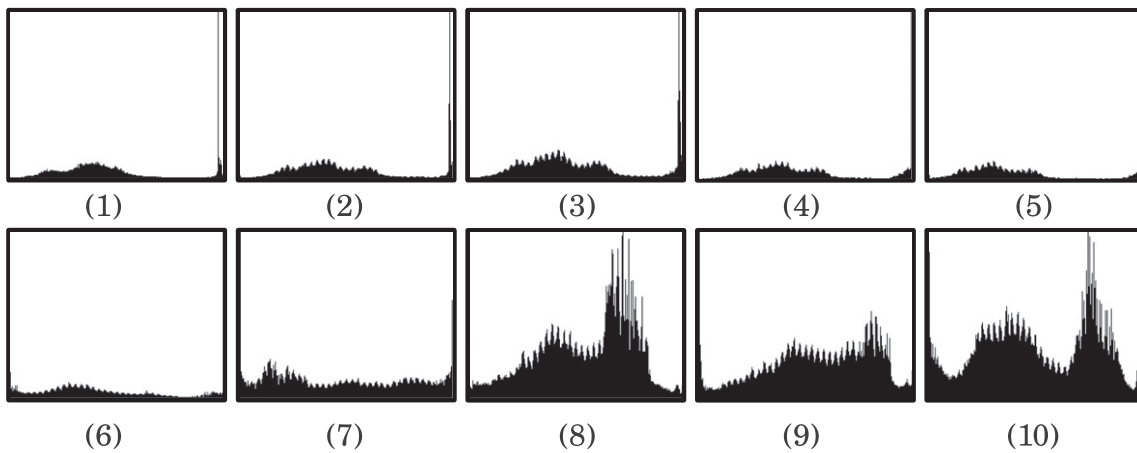


Fig. 9. Set of histograms that represent the brightness and contrast values of each one of the test scenes. The higher the values in the Y axis, the higher the illumination of the scene. The first six scenes correspond to a cloudy day where the objects lack of marked shadows while the remaining four have better illumination conditions.

ducted experiments, both vehicles and pedestrians use to drive along and walk around, respectively, at a normal speed. If this is not the case, most of anomalous behaviours are detected. Fig. 10 shows a sequence of three frames in which two graphical samples of normal and abnormal behaviours are represented. The first two rows correspond to an anomalous behaviour whereas the behavior depicted in the last two rows is normal. As shown in the sequence of frames, the displacement made by a car in the first case is much larger than the second one.

Since this normality component is based on the displacement of the ellipse that wraps the monitored object, the partial occlusions are important to be taken into account, which might cause abrupt changes in the ellipse position and, therefore, in the speed estimation. In other words, while a partial occlusion takes place, the ellipse maintains the visible part of the object and might change the position abruptly when the previous invisible part becomes visible. In the test environment, this phenomenon happens three times (see Fig. 7): (i) lower left part due to the tree branches, (ii)

Table 10

Results obtained by the normality component of speed analysis. Labels C1 (object misclassification), C2 (abrupt position change or wrong size ellipse), C3 (wrong fired rules), and C4 (ellipse swap between objects of different class) refer to error causes.

Scene	Number of frames	Number of situations		Success	Errors			
		Normal	Abnormal		C1	C2	C3	C4
<i>Speed analysis (component 2)</i>								
1	3943	7224	275	7478 (99%)	0	0	21	0
2	3000	7142	137	6923 (98%)	40	0	25	17
3	3000	6751	240	6842 (97%)	0	25	124	0
4	3250	11,051	121	11,095 (99%)	0	0	77	0
5	995	1398	0	1398 (100%)	0	0	0	0
6	1471	2343	53	2344 (97%)	13	39	0	0
7	5233	7831	35	7813 (99%)	0	18	35	0
8	1906	2738	215	2943 (99%)	0	0	10	0
9	2165	6390	299	6639 (99%)	0	15	10	25
10	772	1843	75	1918 (100%)	0	0	0	0
Total				98.7%				

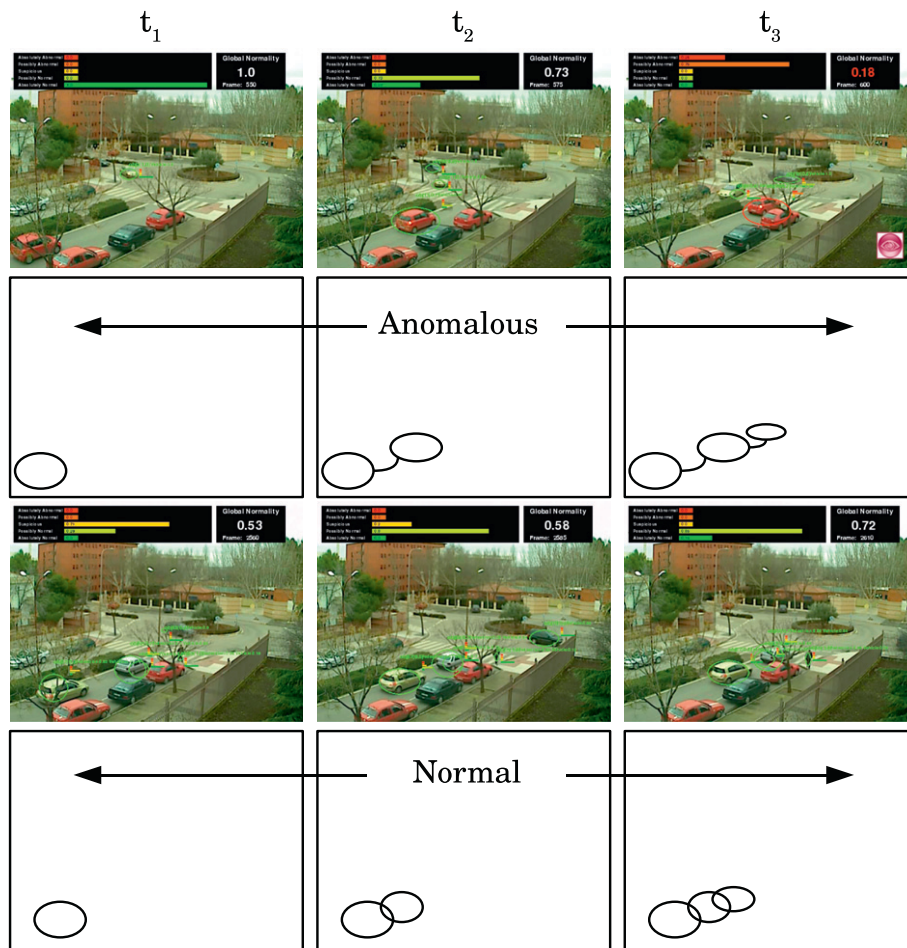


Fig. 10. Graphical examples of a normal and abnormal behaviour according to the speed concept.

upper central part due to the bushes of the roundabout and (iii) right central part due to the fence of the building. The used learning algorithm deals with this problem since it learns that the displacements in this kind of regions may be higher, considering the speed of an object as normal.

Table 11 shows the existing relationship between TP, TN, FP, and FN. The good results are a consequence of the reduced number of false positives and false negatives and the high number of true positives and true negatives.

Tables 12 and 13 expose the results of the global analysis process, making use of the OWA aggregation operators to combine the output given by the two normality components and get a global degree of normality. The errors in this process can be due to the first component, the second or either both at the same time (this last situation is counted as a single error in the global analysis). Occasionally, the errors caused by one of the components do not produce an error in the global analysis due to the use of a vector of weights by the aggregation operator. For instance, a situation

Table 11
Existing relationships between true/false positives and true/false negatives in the normality component of speed analysis.

Scene	TP	TN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR
<i>Coefficient calculation for the component 2</i>											
1	7460	18	21	0	1	0.53	0.99	0.462	0.99	1	0.002
2	6786	137	0	82	0.99	0	0.988	1	1	0.626	0
3	6641	201	39	110	0.984	0.163	0.979	0.838	0.994	0.646	0.005
4	10993	102	15	62	0.994	0.128	0.993	0.872	0.999	0.622	0.001
5	1398	0	0	0	1	–	1	–	1	–	0
6	2344	0	39	13	0.994	1	0.978	0	0.984	0	0.02
7	7788	25	10	43	0.995	0.286	0.995	0.714	0.999	0.368	0.001
8	2918	25	10	0	1	0.286	0.997	0.714	0.997	1	0
9	6589	50	10	40	0.994	0.167	0.99	0.83	0.998	0.556	0.001
10	1843	75	0	0	1	0	1	1	1	1	0

Table 12
Results obtained by the global module combining the analysis of the two normality components. Labels C1 and C2 refer to errors caused by the first and second components, respectively.

Scene	Number of frames	Number of situations		Success	Errors		
		Normal	Abnormal		C1	C2	Both
<i>Global analysis: trajectories and speed</i>							
1	3943	7224	275	7072 (94%)	406	21	0
2	3000	7142	137	6805 (97%)	118	42	40
3	3000	6751	240	6504 (93%)	260	149	0
4	3250	11051	121	10949 (98%)	146	77	0
5	995	1398	0	1397 (99%)	1	0	0
6	1471	2343	53	2190 (91%)	154	39	13
7	5233	7831	35	7537 (95%)	276	53	0
8	1906	2738	215	2910 (98%)	33	10	0
9	2165	6390	299	6370 (95%)	269	50	0
10	772	1843	75	1875 (97%)	43	0	0
Total				95.7%			

Table 13
Existing relationships between true/false positives and true/false negatives in the global normality analysis process.

Sc.	TP	TN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR
<i>Coefficient calculation for the global analysis</i>											
1	6889	183	92	335	0.95	0.33	0.94	0.67	0.98	0.353	0.01
2	6668	137	0	200	0.97	0	0.971	1	1	0.407	0
3	6334	248	44	365	0.946	0.151	0.941	0.849	0.993	0.405	0.01
4	10847	102	19	204	0.982	0.157	0.98	0.843	0.998	0.333	0.001
5	1397	0	0	1	0.99	–	0.99	–	1	0	0
6	2189	14	39	154	0.934	0.736	0.919	0.264	0.982	0.083	0.02
7	7512	25	10	319	0.959	0.286	0.958	0.714	0.999	0.073	0.001
8	2731	179	36	7	0.997	0.167	0.985	0.833	0.987	0.962	0.01
9	6081	289	10	309	0.952	0.033	0.95	0.967	0.998	0.483	0.001
10	1830	75	0	43	0.977	0	0.978	1	1	0.636	0

considered as suspicious by the first component and absolutely normal by the second one might be considered as possibly normal by the global analysis module.

On the other hand, Fig. 11 graphically illustrates the relationship between two of the parameters related to anomalous situations, TN and FP. The number of true negatives (TN) represents the number of anomalous situations detected and correctly understood by the system, while the number of false positives (FP) refers to the anomalous situations unnoticed by the system. The higher the number of false positives and the lower the number of true negatives, then the lower the quality and the efficiency of the methods employed to understand events and situations. The diagrams of Fig. 11 reflect that the proposed methods in this work do not only detect the normal situations correctly but also the anomalous situations that rarely take place.

Tables 14 and 15 show the total and average times spent by each one of the processes that compose the surveillance system. As can be seen, the proposed methods are fast enough to work in real time.

The results obtained prove that the design of components by means of the formal model discussed in this work is feasible, gives a high performance, offers response times really short and, finally, allows to represent knowledge with high interpretability. Furthermore, the two components developed have been successfully combined, thanks to the use of OWA operators, to get a global evaluation of the objects behaviour, which is normal if they follow one normal trajectory at a suitable speed. Although the success rates are often high (between 91% and 99%), it is important to take into account that the duration of the test videos ranges from 30 to 180 s. This fact implies the inclusion of future modifications to

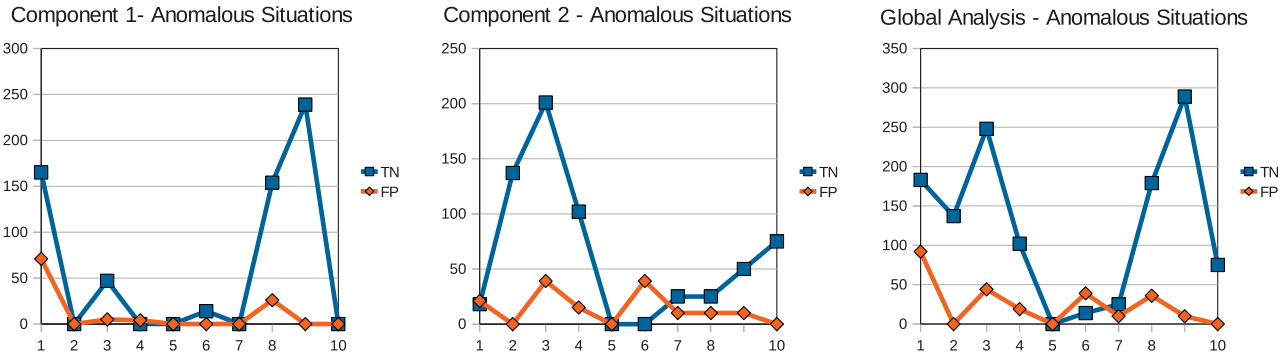


Fig. 11. Relationships between the anomalous situations detected by the normality components and the global analysis. True negatives (TN) represent detected anomalous situations, while false positives (FP) represent anomalous situations understood as normal.

Table 14

Time spent by the system processes, measured in milliseconds. *Sc.* is scene, *elem.* are elements to analyse, *T* is the total time spent, and *M* is the average time.

Scene	Elem.	Object classification		Trajectory analysis		Speed analysis		Global analysis	
		<i>T</i>	<i>M</i>	<i>T</i>	<i>M</i>	<i>T</i>	<i>M</i>	<i>T</i>	<i>M</i>
<i>Time (ms.)</i>									
1	7499	255.86	0.03	113.54	0.02	87.57	0.01	78.75	0.01
2	7005	198.5	0.02	141.93	0.02	78.2	0.01	61.29	0.008
3	6991	214.94	0.03	235.01	0.03	96.84	0.01	79.1	0.01
4	11172	376.16	0.03	272.14	0.02	99.06	0.008	80.21	0.007
5	1398	54.77	0.03	74.33	0.05	14.96	0.01	50.77	0.03
6	2396	68.79	0.02	125.02	0.05	25.31	0.01	39.79	0.01
7	7866	288.05	0.03	183.93	0.02	82.99	0.01	85.68	0.01
8	2953	79.76	0.02	133.56	0.04	26.44	0.008	37.92	0.01
9	6689	221.45	0.03	176.43	0.02	48.33	0.007	73.26	0.01
10	1918	70.41	0.03	63.7	0.03	29.93	0.01	37.88	0.01

Table 15

Time spent by the learning process of fuzzy rules for object and speed classification.

Process	Examples of the training set	Time spent in learning (ms)	Number of fired rules
<i>Time spent by the learning process</i>			
Learning of rules for object classification	4880	1921	56
Learning of rules for speed classification	2680	1610	15

reduce the number of alarm activations, specifically in longer analysis times.

6. Conclusions

This paper presents a novel method to classify the speed of moving objects as normal or abnormal, by analysing the visual information of 2D images gathered by security cameras for surveillance purposes. This method is encapsulated into a normality component within the OCULUS surveillance system. When this component is activated and particularised for a specific monitored environment, the normal behaviour of an object depends on its speed, among other factors. The particularisation of the component is possible thanks to the use of a machine learning algorithm, which generates a set of general and interpretable fuzzy rules. Precisely, these rules classify the speed of moving objects as normal or abnormal. The proposed method offers the following advantages over other existing methods:

- High interpretability of the acquired knowledge.
- Reusability for different environments and kinds of objects.
- Reduced number of samples to build the training set (just one sample per situation).
- High scalability to easily include new kind of constraints in order to apply the component definition.
- Capability of monitoring in real-time.
- Low cost solution.
- No need of previously knowing the camera parameters.

On the other hand, the process of creating training sets for the learning algorithm proposed in this work should be automated to ease the deployment of the surveillance system in new environments. As a possible solution, we have started to design clustering algorithms to address this problem. A first approach has been proposed in Solana-Cipres, Albusac, Castro-Schez, and Rodriguez-Benitez (2009).

Other possible future research line is related to fusion information in low levels. In this work, the output of each normality component is combined to obtain a global normality value. However, fusion processes in low levels can improve the understanding results. In this way, a component may deal with the information gathered by multiple sensors about a particular object. The management of redundant information may also improve the classification and understanding results when the data provided by a single source is not reliable.

Acknowledgments

This work has been funded by the Spanish Ministry of Industry (MITYC) under Research Project HESPERIA (CENIT Project), and the

Council of Science and Technology of Castilla-La Mancha under Projects PII2I09-0052-3440 and PII1C09-0137-6488.

References

- Albusac, J., Castro-Schez, J. J., Lopez-Lopez, L. M., Vallejo, D., & Jimenez-Linares, L. (2009). A supervised learning approach to automate the acquisition of knowledge in surveillance systems. *Journal of Signal Processing*, 89(12), 2400–2414.
- Albusac, J., Vallejo, D., Jimenez-Linares, L., Castro-Schez, J. J., & Rodriguez-Benitez, L. (2009). Intelligent surveillance based on normality analysis to detect abnormal behaviors. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(7), 1223–1244.
- Beymer, D., McLauchlan, P., Coifman, B., & Malik, J. (1997). A real-time computer vision system for measuring traffic parameters. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 495–501).
- Blauensteiner, P., & Kampel, M. (2004). Visual surveillance of an airports apron—an overview of the AVITRACK project. In *Digital imaging in media and education, annual workshop of AAPR*.
- Bloisi, D., Iocchi, L., Remagnino, P., & Monekosso, N. (in press). ARGOS – A video surveillance system for boat traffic monitoring in Venice. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*.
- Castro, J. L., Castro-Schez, J. J., & Zurita, J. M. (1999). Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, 101(3), 331–342.
- Cathey, F. W., & Dailey, D. J. (2005). In *Proceedings of IEEE intelligent vehicles symposium* (pp. 777–782).
- Cheung, K., Kanade, T., Bouguet, J., & Holler, M. (2000). A real time system for robust 3D voxel reconstruction of human motions. In *IEEE computer society conference on computer vision and pattern recognition* (Vol. 2, pp. 714–720).
- Cho, Y., & Rice, J. (2006). Estimating velocity fields on a freeway from low-resolution videos. *IEEE Transactions on Intelligent Transportation Systems*, 7(4), 463–469.
- Dee, H., & Velastin, S. A. (2008). How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 5(19), 329–343.
- Fawcett, T. (2003). *Roc graphs: Notes and practical considerations for data mining researchers*. Technical report hpl-2003-4. HP Laboratories, Palo Alto, CA, USA.
- Forsyth, D. A., & Ponce, J. (2002). *Computer vision: A modern approach*. Prentice Hall Professional Technical Reference.
- Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W⁴: Real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence*, 22(8), 809–830.
- Hu, Z., Wang, C., & Uchimura, K. (2008). 3D Vehicle extraction and tracking from multiple viewpoints for traffic monitoring by using probability fusion map. *Electronic Letters on Computer Vision and Image Analysis*, 7(2), 110–119.
- Lee, L., Romano, R., & Stein, G. (2000). Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 758–767.
- Liu, W., Yamazaki, F., Vu T. T., & Maruyama, Y. (2007). Speed detection of vehicles from aerial photographs. In *Proceedings of 27th Asian conference on remote sensing* (6p).
- Maduro, C., Batista, K., Peixoto, P., & Batista, K. (2008). Estimation of vehicle velocity and traffic intensity using rectified images. In *15th IEEE international conference on image processing* (pp. 777–780).
- Magee, D. R. (2004). Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22(2), 143–155.
- Palaio, H., Maduro, C., Batista, K., & Batista, J. (2009). Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking. In *IEEE international conference on robotics and automation*.
- Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach*. Pearson Education.
- Solana-Cipres, C. J., Albusac, J., Castro-Schez, J. J., & Rodriguez-Benitez, L. (2009). Automatic object labelling for monitored environments using clustering techniques. In *Proceedings of the 3rd international conference on imaging for crime detection and prevention*.
- Smith, G. (2004). Behind the screens: Examining constructions of deviance and informal practices among CCTV control room operators in the UK. *Surveillance and Society*, 2(2), 376–395.
- Valera, M., & Velastin, S. A. (2005). Intelligent distributed surveillance systems: A review. *Vision, Image and Signal Processing*, 2(152), 192–204.
- Vallejo, D., Albusac, J., Jimenez-Linares, L., Gonzalez-Morcillo, C., & Moreno, J. (2009). A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert System with Applications*, 46(3), 10503–10511.
- Velastin, S. A., Khoudour, L., Lo, B., Sun, J., & Vicencio-Silva, M. (2004). PRISMATICA: A multi-sensor surveillance system for public transport networks. In *12th IEEE international conference on road transport information and control* (pp. 19–25).
- Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1), 183–190.
- Zadeh, L. A. (1965). *Fuzzy Sets, Information and Control*, 8(3), 338–353.
- Zadeh, L. A. (2001). From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. *Annals of the New York Academy of Sciences*, 929, 221.