Advanced visual-based traffic monitoring systems for increasing safety in road transportation

G.L. Foresti C. Micheloni L. Snidaro

Department of Mathematics and Computer Science (DIMI)
University of Udine, Via Delle Scienze 208 – 33100 Udine (Italy)
e-mail: {foresti,michelon,snidaro}@dimi.uniud.it

Abstract

Automatic traffic monitoring systems play an important role in the process of increasing the safety of humans and vehicles in road transportation. Vision-based approaches are promising since they allow the control of larger detection areas and the reduction of installation and maintenance costs with respect to other sensors like inductive loops which require pavement reconstruction. However, image processing techniques show robustness problems due to variable light conditions and real-time constraints required by a traffic monitoring system. An exhaustive survey of visual-based traffic monitoring systems and an overview of the principal computer vision techniques normally applied in these systems will be shown.

Keywords – Traffic monitoring systems, computer vision, image processing, vehicle detection

1. Introduction

Road transportation systems have been subjected to considerable increases in congestion and accidents during the last decade. A direct consequence of this condition has been a reduction in vehicular and human safety. In Italy, there are about 7 million of vehicles moving every day on urban, extra urban roads, and highways: every year about 200.000 accidents that cause about 6500 fatalities and 220.000 injuries must be noticed. In Europe road accidents are estimated to cost about 45 billion euros per year, consisting in 15 billion for medical care, police involvement and vehicle repairs and 30 billion in lost economic production due to fatalities or injuries. With 45 000 victims annually, the avoidance of a fatal accident would imply saving 1 million Euros. The U.S. National Highway Traffic Safety Administration reports that in the 1999 the total economic cost of motor vehicle crashes was $230.6 billion. This represents the present value of lifetime costs for 41821 fatalities, 5.3 million non-fatal injuries, and 28 million damaged vehicles, in both police-reported and unreported crashes [1], [7].

In order to reduce both traffic accidents and congestion, the enhancement of traffic monitoring [3], [8], [16], [24], [36], [68] and driving assistance systems [5], [14], [23], [26], [57] become important issues. Improved safety can be achieved by (a) providing a human operator in a remote control room with real-time information (acquired on the basis of current conditions of the environment) about the situation of the traffic flow or accidents, (b) helping the human driver with automatic driving assistance procedures such as keeping a safe distance from the ahead vehicle, obstacle detection in bad visibility conditions, etc.

Important components of a real-time advanced traffic monitoring system are the acquisition, processing, and interpretation of the available sensory information regarding the traffic conditions. At the lowest level, sensory information is used to derive discrete signals to drive the monitoring system and at a higher level this information is used to study the patterns of traffic flow or to adjust the behaviour of global traffic parameters. Information about the traffic status may come from a variety of sensors such as visual (optical, and infrared), radar, laser and ultrasonic range-finders, global positioning systems (GPS), loop or pneumatic detectors, microwave sensors. Such information can be used to detect, localize and track vehicles, detect possible accidents, detect dangerous behaviours of some vehicles (abnormal trajectory, unsafe distance from other vehicles, etc.). Among the set of available sensors, the most commonly used is the loop detector. However, loop detectors present significant errors in their measurements and are very expensive due to high costs of installation and maintenance.

The new generation of visual sensors (e.g., CCD colour progressive cameras) and related computer vision hardware provide information that is richer and more complete than other sensors, carrying many researchers and industries to start the development of traffic flow monitoring systems based on computer vision.

Visual sensors take a fundamental role also in the development of automatic driving assistance systems as they are able to provide different information on relatively large regions. Visual data taken from a vehicle may be used to detect and track obstacles (e.g., pedestrians, vehicles, etc.) and keep a safe distance from them. In addition, knowledge about surface features on the obstacles or about their shape may further improve the robustness of the driving assistance system. A detailed review of these systems can be found in [4], [6], [24], [37], [45], [61].

This paper will focus on real-time visual-based systems for traffic monitoring. In Section II, a survey of the state of the art on traffic monitoring systems equipped with visual sensors will be presented. Then, in Section III the general architecture of a visual-based traffic monitoring system will be discussed. Section IV will present the change detection techniques commonly applied to find moving objects in outdoor image sequences. Section V and VI will describe two of the most important modules of a visual-based traffic monitoring system: the vehicle recognition and the vehicle tracking modules. Finally, Section VII will describe the procedures for traffic flow analysis from visual data.

2. State of the art on traffic monitoring systems

Although several systems are currently in progress, the behaviour of vision systems for traffic monitoring is still not satisfactory. Generally, these systems are limited to measure or quantify the traffic flow, to solve partially problems such as congestion detection in highways [10], [25], queue detection [63], etc. Few systems have considered the problem of providing detailed information about the behaviour of individual vehicles in order to estimate and/or predict possible dangerous situations. As indicated in the work of Zhu et al. [68], a successful and widely used vision system for real traffic monitoring applications must meet the following four basic requirements:

(a) *easy installation and calibration* - this is important for on-site set-up, reconfiguration and operation by non-expert personnel.

(b) *environmental adaptation* - the real system should work in different light conditions, including heavy shadows under strong sunlight, reduced illumination in the evening, vehicle headlights at night and abrupt light changes.

(c) *accurate vehicle speed and size estimation* – it is needed for applications such as intersection control, traffic surveillance, speed trap detection, vehicle classification and other special studies.

(d) *real-time operation and low cost* - these are a key factors for the wide use of an on-line traffic monitoring system.

In the last years, many works have proposed computer vision techniques for traffic monitoring and vehicle control in road or highway systems. D'Agostino and Shuldiner [12] discussed the potentials of a commercial machine vision system for traffic monitoring and control. The basic requirements are low cost and robust performance, which have not been completely met till now. The system presented by D'Agostino and Shuldiner is unable to work during night time, it needs expensive special image processing hardware and it shows problems in presence of shadows. Houghton et al. have proposed a system for tracking vehicles in video-images at road junction [30]. Inigo has presented a machine vision for traffic monitoring and control [32]. Michalopoulos [44] has developed the Autoscope system for vision-based vehicle detection and traffic flow measurement. This system computes parameters such as average vehicle speed and spatial occupancy.

None of the previous works considers the problems of difficult illumination conditions or fast changing environment conditions. Kilger has done extensive work on shadow handling in a video-based, real-time traffic monitoring system [36]. Shadows are separated from vehicles by investigating an edge image of detected regions. The speed of a vehicle is estimated by tracking the middle point of the vehicle's front edge in the image sequence, where a constant speed assumption is made. Main limitations of this system are the lack of speed estimation results and the absence of 3D measurements of vehicles. Zielke et al. [69] proposed a method for detecting and tracking cars based on symmetry. This method can be used in situations where a camera is mounted on another vehicle in the same lane. However, this is not the best viewing position for monitoring traffic. A fully symmetrical view of a vehicle within an image is possible only from some particular vantage points.

In other systems, vehicle shapes are modelled using complex models [62],[67],[34],[15], which cannot be processed in real-time with low-cost hardware. Sullivan [62] used a wire-frame model for vehicles. Based on an estimated pose and calibration of both intrinsic and extrinsic parameters of the camera, the model is back-projected to the image and edges of the model are then matched to edges in the image. The system shows problems if the estimated pose of the vehicle is not accurate. Yuan et al. [67] proposed a system able to detect a vehicle moving on a road and estimate its length, width, height to classify it into many categories. This approach is affected by the general problems in image segmentation, and the methods proposed to identify the different subparts of the vehicle are quite ad hoc. More recently Jolly et al. [34] presented a system able to extract vehicles and their wheels by using a sophisticated image segmentation method and a deformable model of the vehicle. This method presents some limitations in the real-time functioning and in the presence of occlusions. The hypothesis that the entire vehicle is fully visible in a single image is assumed.

All of these approaches require the use of high cost hardware for a real-time implementation and assume the entire vehicle to be fully visible in a single image, which is not always true.

Ferrier et al. [15] obtained real-time performance (on a SUN IPX with a Datacell S2200 image capture board) by tracking the occluding contour using intensity/motion information. Initial calibration of a projection relationship between an image and the ground plane enable metric information to be derived from the image positions and velocities without full calibration. The main drawback of the paper is that the tracked outline of the vehicle is roughly plane shape and this weak perspective viewing condition can only be satisfied if the camera is far enough above the road being viewed. Effects of headlights at night were not discussed in the paper.

Recently, Smith et al. [56] proposed robust detection and tracking techniques for intelligent vehicle-highway applications where computer vision plays a fundamental role. In particular, an adaptive filtering scheme is used to track feature windows on the target in spite of the unconstrained motion of the target, possible occlusions of feature windows, and changing target and environmental conditions. Relatively high-speed targets are tracked under varying conditions with operating parameter estimates and no explicit target models. Experiments are focused on the tracking of a limited number of vehicles. Moreover, the vehicle tracking is performed from a visual system placed on-board one vehicle and looking to obstacles in front of it. Zhu et al. [68] presented a new approach for automatic traffic monitoring using 2D spatio-temporal images. A TV camera is placed above the highway to monitor the traffic through two slice windows for each traffic lane. One slice window is a detection line perpendicular to the lane and the other is a tracking line along the lane. The problems of vehicle counting, speed estimation and vehicle classification have been addressed. In particular, the vehicle classification is performed by using 3D measurements (length, width and height). The system, called VISATRAM (vision system for automatic traffic monitoring), has been tested with real road images under various light conditions, including shadows in daytime and lights at night.

3. General architecture of a visual-based traffic monitoring system

Figure 1 shows the general architecture of a visual-based traffic monitoring system. A set of fixed or mobile (in pan and tilt) CCD cameras is generally used to acquire image sequences of the monitored road.

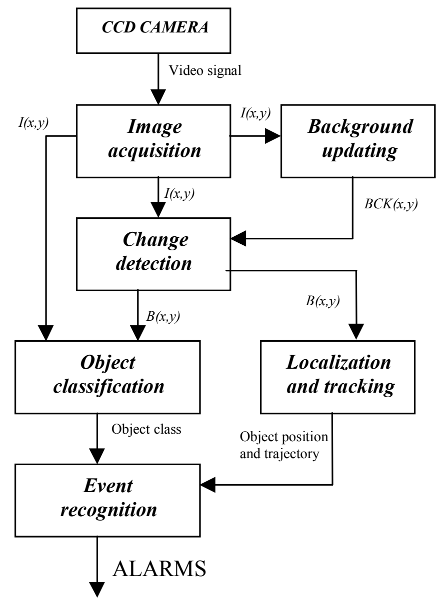

Fig. 1 – General architecture of a visual-based traffic monitoring system

Figure 2a shows a running example representing a highway with a bi-directional lane. Multiple vehicles are moving in the considered sequence. In order to build an advanced vision system able to robustly track traffic objects in unpredictable, real-world conditions, the system is required to have means to detect such objects automatically. A change detection module (CD) [53] is generally applied. It considers the input image as a set of pixels that belong to one of two categories: moving objects or background. Moving pixels are those which are believed to belong to a traffic object of interest, while background pixels belong to the environment. The CD module compares each input frame $I(x,y)$ with a background image (Figure 2c) which represents the monitored scene without moving objects. This procedure yields a binary image $B(x,y)$ where each pixel is classified as a moving point or background. Groups of connected pixels (commonly called blobs) belonging to the class of moving points represent possible objects (e.g., a vehicle, a pedestrian, etc.) moving in the scene. Figure 2b shows the blobs extracted from the input image sequence in Figure 2a. A background updating procedure based on the Kalman filter is generally applied to estimate significant illumination changes on the background scene [18].

A localization and tracking module, which uses as input geometric information about the blobs, is necessary for estimating the motion parameters of detected objects [17], [18], [19], [33]. A camera calibration procedure [39],[64] and the ground-plane hypothesis [62] are considered to solve the ill-posed problem of transforming points from the image plane to a 3D general reference system of the scene. In several systems, a 2D top view map of the monitored scene is used to display to a remote operator the motion parameters of the detected vehicles, e.g., position, speed, motion direction, etc. The vehicle position, speed and direction can be represented by the position, orientation and magnitude of a 2D vector displayed on the 2D top view map.

Figure 2d shows an example of the map. Finally, the high level part of the traffic monitoring system is generally organized to address two complex problems: (a) to classify each detected object among a predefined set of models (cars, motorcycles, vans, lorries, buses, pedestrian, etc.) [20] and (b) to understand whether the behaviour of these objects is normal or potentially dangerous (non linear trajectories, crossings of the road in non authorized areas, etc.) [9], [22], [42], [43], [66].

4. Change detection

In the literature, change detection (CD) is performed generally at the pixel level, at the edge level or at higher feature levels (e.g., lines, corners, etc.) [58]. CD algorithms at feature levels require more computational efforts with respect to those working at pixel level, but they provide more accurate information for higher system modules. In this paper, the focus is on real-time processing methods, so CD at the edge level or at higher feature levels are not considered. Additional information on these techniques can be found in [53],[59].

Several change detection methods have been developed in the last years. Let $I(x,y)$ be a $N\times N$ input image. The simple difference (SD) method [51] computes the absolute difference $D(x,y) = |I_1(x,y) - I_2(x,y)|$ between the pixel intensities of the input images, and thresholds the resulting difference image:

$$B(x,y) = \begin{cases} 0 & if \quad D(x,y) < Th \\ 1 & otherwise \end{cases} \qquad 0 \le Th \le 255 \tag{1}$$

where Th is some arbitrary threshold depending on the illumination degree of the scene. This method is the faster, but results are very noise dependent.

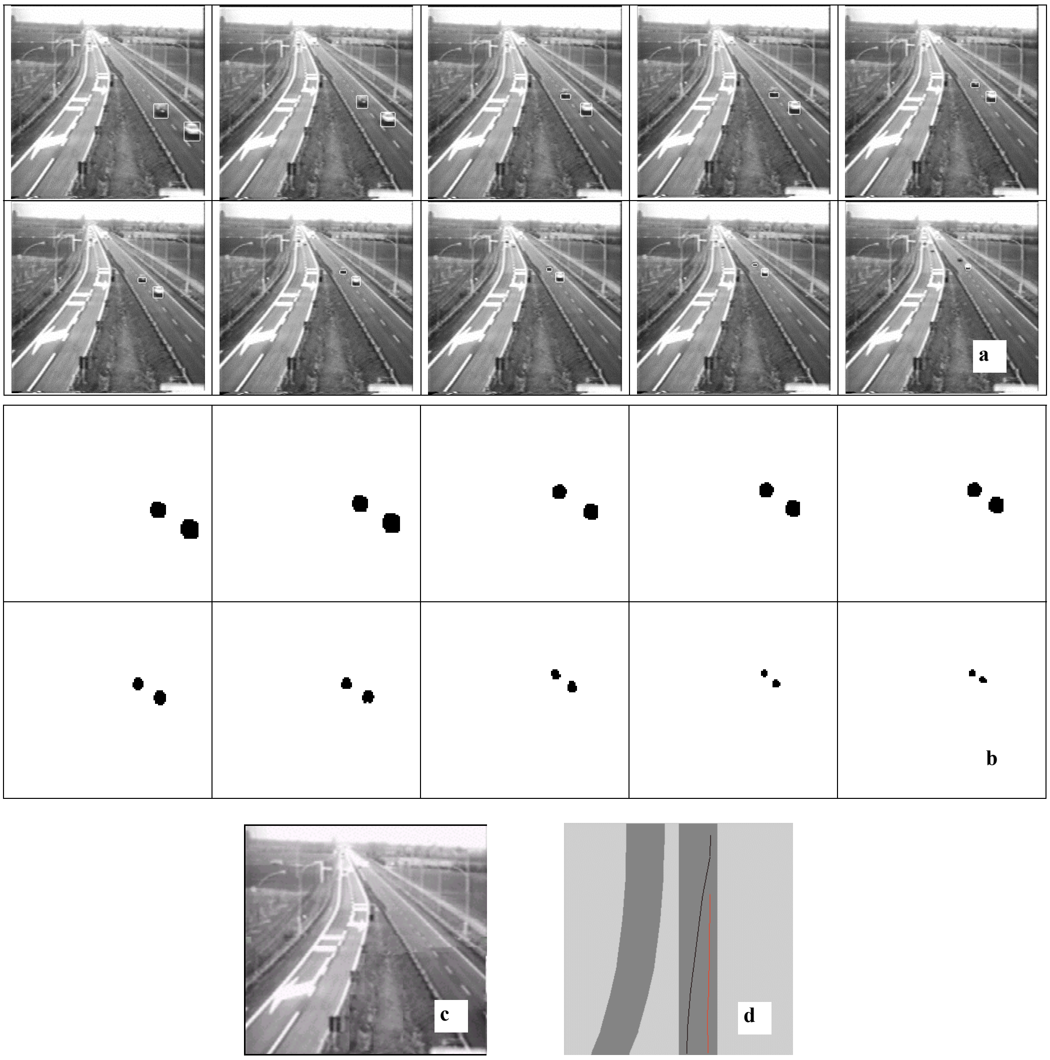

Fig. 2 – (a) An image sequence acquired on a highway with a bi-directional lane equipped with a visual-based traffic monitoring system, (b) the related blob images, (c) the background image and (d) a 2D top view map of the monitored area

To overcome the problem of sensitivity to noise, the geo-pixel (GP) method [58] considers *nxn* pixel regions in the two input images, and computed a likelihood ratio L_{ij} by using the means and the variances of the two regions R_i and R_j. The output binary image is obtained as:

$$B(x,y) = \begin{cases} 0 & \text{if } L_{ij} < L_{Th} \\ 1 & \text{otherwise} \end{cases} \quad \forall (x,y) \in R_i, R_j;\ i,j = 1,..,\left\lfloor \frac{N}{n} \right\rfloor \tag{2}$$

where L_{Th} is a threshold derived from experiments.

The shading method (SM) [58] models the intensity at a given point $I_p(x,y)$ as the product of the illumination $I_i(x,y)$ and a shading coefficient S_p which is calculated for each point as $I_p=I_i \cdot S_p$. The Phong's shading model is used [48]. It can be easily proved that, to establish if a change has taken place in a given region R_i over two consecutive frames, $I_j(x,y)$ and $I_{j+1}(x,y)$, it is sufficient to calculate the variance σ_i of the intensity ratios I_{j+1}/I_j in that region. If si is close to zero, no change is perceived to have taken place.

Recently, more sophisticated CD approaches have been presented. The LIG method [46] is based on the assumption that pixels at locations having high gray level gradient form a part of an object, and that nearly pixels with similar gray levels will be also part of the same object. The intensity gradient is computed as:

$$G(x,y) = \min\{I(x,y) - I(x \pm 1, y \pm 1)\} \tag{3}$$

The $G(x,y)$ image is unidirectional and gives large negative gradients at pixels that are on the object boundaries. To this end, the $G(x,y)$ image is divided into *m*x*m* sub-images in order to limit the effects of illumination changing on the computation of local means and deviations. The regional means and deviations are first smoothed using the neighbouring regions and then interpolated to fill a *m*x*m* region again. Finally, a threshold procedure is applied to isolate object pixels from the background.

In the work of Stauffer and Grimson [60], motion detection is performed though an adaptive background subtraction method that models every pixel of the background as a mixture of Gaussians and uses an on-line approximation to update the model.

The recent history of each pixel, $\{X_1, \ldots, X_t\}$, is modelled as a mixture of *K* Gaussians distributions. The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{4}$$

where *K* is the number of distributions, and $\omega_{i,t}$, $\mu_{i,t}$, $\Sigma_{i,t}$ are the weight, the mean, and the covariance matrix of the *i*-th Gaussian of the mixture at time *t*. The weight coefficient accounts for the amount of data represented by the Gaussian. η is a Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{n/2}} e^{-1/2(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \tag{5}$$

Every new pixel value is checked against the existing *K* distributions, if the value is within 2.5 standard deviations of a distribution then the pixel matches that distribution.

To guarantee low computational requirements, the mixture model is updated through an on-line *K*-means approximation of the expectation maximization (EM) algorithm.

A very good survey on these algorithms can be found in [53]. The problem of automatic thresholding was recently investigated also by Snidaro and Foresti [59].

Durucan and Ebrahimi [13] proposed in the Wronskian Change Detector. This approach exploits the concept of linear dependence and independence between vectors as a means for determining changed regions.

Ziliani and Cavallaro [70] compare the statistical behaviour of a small neighbourhood, *N*, of each pixel position *k* in the squared difference image. A significance test, based on a threshold α, is used to compare the statistics in N with a model M, and to decide whether the pixel k belongs to a changed area or to the background. M is described by a χ^2 distribution and the significance threshold α is a stable parameter that should be set to values close to zero (i.e. 10^{-2} to 10^{-6}).

5. Object recognition

Object recognition represents one of the main modules of a visual-based traffic monitoring system. It receives in input feature-based information (shape descriptors [47]) on the detected objects and produces as output the classes to which these objects belong. Each class corresponds to one of the predefined object models stored into a database.

Shape descriptors are feature vectors extracted from image patterns. They can be used by classification systems to synthesize the morphological properties of objects, to identify them or to transmit their appearances. If the goal is to assign an observed pattern to a predefined class, it is important to obtain the information about the shape aspects that allow one to differentiate an object from other objects. A wide class of shape descriptors has been proposed in the literature and an exhaustive survey has been proposed by Pavlidis [47] and more recently by Loncaric [38]. An important class of shape descriptors is based on Mathematical Morphology which transforms images progressively by applying morphological operators and by analyzing and keeping track of the loss of information which occurs by successive steps [2],[40],[41],[54].

The most widely applied class of shape descriptors is represented by invariant moments [49]. Let $m_{p,q}$ be the geometric moment of order $(p+q)$:

$$m_{p,q} = \sum_{i=1}^{M}\sum_{j=1}^{N} x_i^p y_j^q B(x_i, y_j) \qquad (p,q = 0,1,2,....) \tag{6a}$$

where $B(x,y)$ represents the blob image. However, this definition of moment of order $(p+q)$ is not invariant to changes in scale, rotation and translation. In order to generate translation invariants, the $(p+q)$-th central moment should be considered:

$$v_{p,q} = \sum_{i=1}^{M}\sum_{j=1}^{N} (x_i - x_0)^p (y_j - y_0)^q B(x_i, y_j) \qquad (p,q = 0,1,2,....) \tag{6b}$$

where (x_0, y_0) represents the center of mass of the blob whose coordinates are defined as $x_0 = m_{1,0}/m_{0,0}$ and $x_0 = m_{0,1}/m_{0,0}$. To introduce scale invariance, the central moment should be normalized with the use of the term introduced by Hu [31]:

$$\mu_{p,q} = \frac{v_{p,q}}{(m_{0,0})^{\beta}} \tag{6c}$$

where $\beta = 1 + (p+q)/2$. By applying the theory of algebraic invariants to the scale normalized invariants (see eq. 4c), it is possible to obtain the second and third order moment invariants which are invariant under rotation as well as translation and scale change (see APPENDIX I).

Other sets of moments have been introduced in the literature. These include Legendre moments, rotational and complex moments, and Zernike moments. A detailed review can be found in [65].

Recognition techniques are mandatory to add to visual-based traffic monitoring systems intelligent capabilities useful to understand the behaviour of dynamic events in complex scenarios. Decision trees (DTs) [55] and artificial neural networks (ANNs) [28] have been used until now for such purposes. DTs are based on information theory, while ANNs rely on specific training algorithms. Both ANNs and DTs learn a concept representation from instances of the concept itself by induction. In particular, they subdivide, during the learning phase, the decision space into regions delimited by hyperplanes: DTs produce hyperplanes orthogonal to the space axis, while ANNs produce more general hyperplanes.

DTs represent a popular approach to pattern recognition [55],[50]. A DT is a recursive structure composed by two kinds of nodes: (a) internal nodes having child nodes, each of which is also a DT, and (b) terminal nodes (*leaf nodes*) having no child nodes. The root of tree divides the feature space into subsets (*splitting*), assigning each subset to each child node. The splitting process is continued until each leaf node corresponds to one class. The main problem of DTs is that it requires an exhaustive search through a list of arbitrarily generated splits to find the best split [55]. However, this process is computationally inefficient and the obtained solution could be not close to the optimal solution.

NNs have solved a wide set of classification problems and are able to work in a parallel way. In the context of traffic monitoring systems NNs with hidden layers, such as a multi-layer perceptron (MPL) [28], are generally used. They are able to work when input data are not linearly separable, but the exact number of hidden neurons and the connectivity between layers must be specified before the training phase. Moreover, it is not guaranteed that correct weights will be found for a given number of neurons and a particular training set. In many cases, the number of hidden neurons is chosen by trials and error minimization.

Recently, Foresti presented a new ANN architecture, called backtracking neural tree (BNT), for classify objects in traffic scenes [20]. This classifier provides a solution for both the main limitations of DTs and NNs with hidden layers. In particular, it overcomes the classical problems of optimum topology definition, selection of the optimum number of hidden layers, selection of the optimum number of neurons per layer, etc. It is built up by means of a large set of patterns which are obtained by extracting from several image prototypes some feature vectors characterizing the considered object models, e.g., cars, lorries, buses, motorcycles, pedestrian, etc. Object features are extracted by means of the approach suggested in [29]. In particular, the minimum bounding rectangle (MBR) related to the detected blob b_i in the $B(x,y)$ image is divided into four parts, q_1, q_2, q_3, q_4, called quartiles, according to the position of the center of mass b=(bx,by) of the blob itself (Figure 3a). Moreover, the four distances between the point b and the center of mass of the four quartils $\boldsymbol{bq}_1$, $\boldsymbol{bq}_2$, $\boldsymbol{bq}_3$, $\boldsymbol{bq}_4$ (Figure 3b) are computed. The four distance values d_1, d_2, d_3, d_4 represent the blob distribution inside the MBR and give a measure of the blob shape. In order to increase the robustness of such a representation, the angles a_1, a_2, a_3, a_4 formed by the vectors, d_i, with the horizontal axis of the 2D image reference system centered in the point **b** have been used as features (Fig. 3c). The pattern $\boldsymbol{p}(b_i)$, related to the detected blob b_i, is composed by the following eight values:

$$\mathbf{p}(b_i) = [d_1, d_2, d_3, d_4, a_1, a_2, a_3, a_4] \qquad (7)$$

In [11], Collins et al. proposed two object classification algorithms. The first uses view dependent visual properties to train a neural network classifier to recognize four classes: single human, human group, vehicles, and clutter. The second method uses linear discriminant analysis to provide a finer distinction between vehicle types (e.g. van, truck, sedan) and colours.

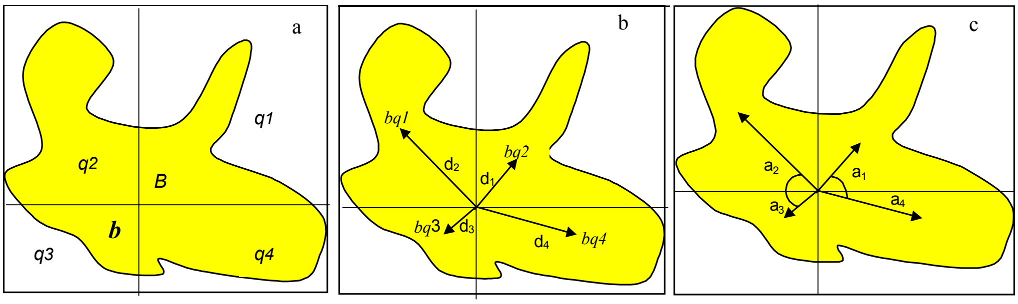

Fig. 3 – (a) Four quartiles for the blob B, (b) the four distances between the point b and the center of mass of the four quartils and (c) the four angles a1, a2, a3, a4

In the first method the neural network is a standard three-layer network, the learning is accomplished using the backpropagation algorithm. Input features to the network are a mixture of image-based and scene-based object parameters: image blob dispersedness (perimeter2/area (pixels)); image blob area (pixels); apparent aspect ratio of the blob bounding box; and camera zoom.

The second method has two sub-modules: one for classifying object "shape", and the other for determining "color". Each sub-module computes an independent discriminant classification space, and calculates the most likely class in that space using a weighted *k*-class nearest-neighbour (*k*-NN) method. To calculate both discriminant spaces, Linear Discriminant Analysis (LDA) calculation is used. It calculates the average covariance matrix of points within each class (W) and between different classes (B), then computes the eigenvalues l and eigenvectors b_i of the separation matrix $W^{-1}B$. The eigenvector b_i associated with each eigenvalue l_i provides the coefficients of the *i*-th discriminant function, which maps feature vector x into a coordinate in discriminant space. During on-line classification, feature vector x is measured for a detected object, and transformed into a point y in discriminant space. To determine the class of the object, the distance from point y to points representing each labelled training example is examined, and the k closest labelled examples are chosen. These are the k nearest neighbours to y. According to the *k*-NN classification rule, the labels of these nearest neighbours provide votes for the label (class) of the new object, and their distance from y provides a weight for each vote. The class of y is chosen as the class that receives the highest weighted vote.

For shape classification, area, center of gravity, and width and height of the motion blob in each sample image are computed . The system also calculates 1st, 2nd and 3rd order image moments of each blob, along the *x*-axis and *y*-axis of the images. Together, these features comprise an 11-dimensional sample vector of calculated image features. Then, the discriminant space for shape classification using the LDA is calculated.For colour classification sampled RGB values are mapped into (I1,I2,I3) colour space values then, the system averages the calculated (I1,I2,I3) values to get a single 3-dimensional colour feature vector for the that image.

In [35], Kang et al. presented a method of grouping and matching line segments to recognize objects. They proposed a dynamic programming-based formulation extracting salient line patterns by defining a robust and stable geometric representation that is based on perceptual organizations. Several junctions are determined within the image and grouped by using the collinear constraint between them. Junction groups similar to the model are searched in the scene, based on local comparison.

6. Object tracking

Object tracking consists in identifying at each frame the position on the image plane of a given object. This procedure requires to find the right match between multiple blobs belonging to two consecutive frames (Figure 4). The matching function should take into account different information such as: (a) center of mass of the blob at the current frame, (b) estimated center of mass of the blob at the next frame, (c) dimensions of the minimum bounding rectangle (MBR) of the detected blob, (d) area of the blob, (e) ratio between the area and the perimeter of the blob, (f) colour histogram information, etc. More details on the matching procedure can be found in [21].

The object tracking module is generally composed by as many linear Kalman filters (KFs) or extended Kalman filters (EKFs) as the number of the extracted blobs from the image. Each Kalman filter is characterized by a state vector $\boldsymbol{x}(k)$ of several elements that should be estimated at each frame: (a) the coordinates (cx_i,cy_i) of the center of mass of the blob b_i on the top view map, (b) the dimensions (l_i,h_i) of the MBR of the blob, (c) the ratio $r=h/l$ between the height and the side of the MBR of the blob. These features are initialized according to the value measured (in pixels) on the blobs detected on the first frame. More details on the KF structure (updating and prediction phases) can be found in [21]. Another recent and effective tracking technique based on a different approach is the Meanshift algorithm; details can be found in [72].

Some constraints can be introduced in order to reduce the number of possible matches: (i) maximum displacement of the object on two consecutive frames (this constraint allows to define a confidence circle in which to search for candidates for the matching), and (b) maximum variation of the value r between two consecutive frames (this allows to consider only objects with a similar size). The following matching algorithm is based on the simple nearest neighbour technique and is here reported for tutorial purposes, other more complex approaches can be found in [71].

Let $B_i(t)$ $i=1,..,n,$ be the set of n blobs detected by the system at the time instant t, and $B_j(t\text{-}1)$ $j=1,..,m$, be the set of m blobs detected at the time instant $(t\text{-}1)$ that satisfy the above constraints. Then, let $(b_i(t).x\ ,\ b_i(t).y)$ be the coordinates in pixels of the real position of the object $b_i(t)$ at the time instant t, and let $(b_j(t\text{-}1).x\ ,\ b_j(t\text{-}1).y)$ be the coordinates in pixels of the real position of a candidate blob. Let now define the matching function as follows:

$$F_{ij}(t)=\sqrt{(b_i(t).x-b_j(t-1).x)^2+(b_i(t).y-b_j(t-1).y)^2} \qquad (7)$$

The best candidate will be the candidate $b_k(t\text{-}1)$ where the index k minimizes the following function:

$$F_{ik}(t)=\min_{j=1,\dots,m} F_{ij}(t) \qquad (8)$$

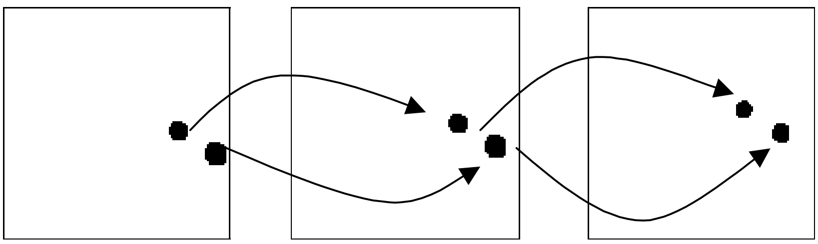

Fig. 4 – Results of the blob matching procedure on the image sequence in Figure 2

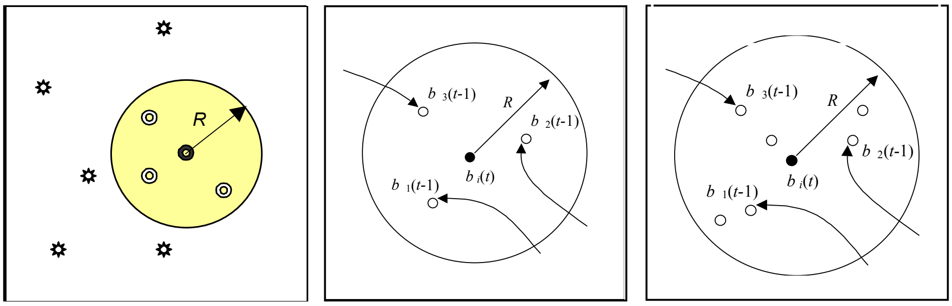

Fig. 5 – (a) Confidence circle of radii R centered on the blob selected for tracking, (b) three candidate blobs found at the same distance from the considered blob and (c) selection of the best match on the basis of the minimum distance from the estimated position of the candidate blobs.

Figure 5 shows a possible situation in which three different candidates at the same distance from the considered blob. In this case, the matching function points out three equal values. In order solve the ambiguity, the estimated position of each object has been considered.

Let ($\hat{b}_j(t).x$, $\hat{b}_j(t).y$) be the coordinates in pixels of the estimated position of the object candidates for the match. These coordinates can be computed by taking into account the position of the object in the last three frames:

$$\hat{b}(t).x = 1/2\{[b(t\text{-}1).x\text{-}\ b(t\text{-}2).x] + [(b(t\text{-}2).x - b(t\text{-}3).x)]\} \quad (9a)$$

$$\hat{b}(t).y = 1/2\{[b(t\text{-}1).y\text{-}\ b(t\text{-}2).y] + [(b(t\text{-}2).y - b(t\text{-}3).y)]\} \quad (9b)$$

Finally, the updated matching function becomes:

$$F_{ij}(t) = \sqrt{(b_i(t).x - b_j(t-1).x)^2 + (b_i(t).y - b_j(t-1).y)^2} + \\ + \sqrt{(b_i(t).x - \hat{b}_j(t).x)^2 + (b_i(t).y - \hat{b}_j(t).y)^2} \quad (9c)$$

Figure 5c shows the same situation described in Figure 5b with the estimated object positions. The blob $b_3(t\text{-}1)$ is the blob that follows with greater accuracy the trajectory of $b_i(t)$.

In real image sequences, occlusions or noise can generate problems for the tracking algorithm. In fact, there are some situations in which objects really present in the scene are not detected by the visual system. In road or highway images, typical cases of occlusions are generated by fast vehicles passing slower vehicles. Since these situations occur very frequently, the problem of occlusions should be considered in designing the tracking module. Different hypothesis should be taken into account. A vehicle can disappear from the scene for two causes: (a) it exits the field of view of the camera or (b) it is occluded by another vehicle. In both cases, the tracking algorithm finds a vehicle in the previous frame that cannot be matched with any vehicle in the current frame. Such a object is labelled as a *ghost* object and it is maintained in the 2D map of the scene (Figure 6) [27]. Its position is updated for a given k number of successive frames according to eq. (8). An object can exit from a ghost state if the following events occur: (a) the object goes outside the borders of the map (it is eliminated), (b) it disappears from the scene for more than k frames (it is eliminated), (c) it is correctly matched with a real object (it appears again in the scene).

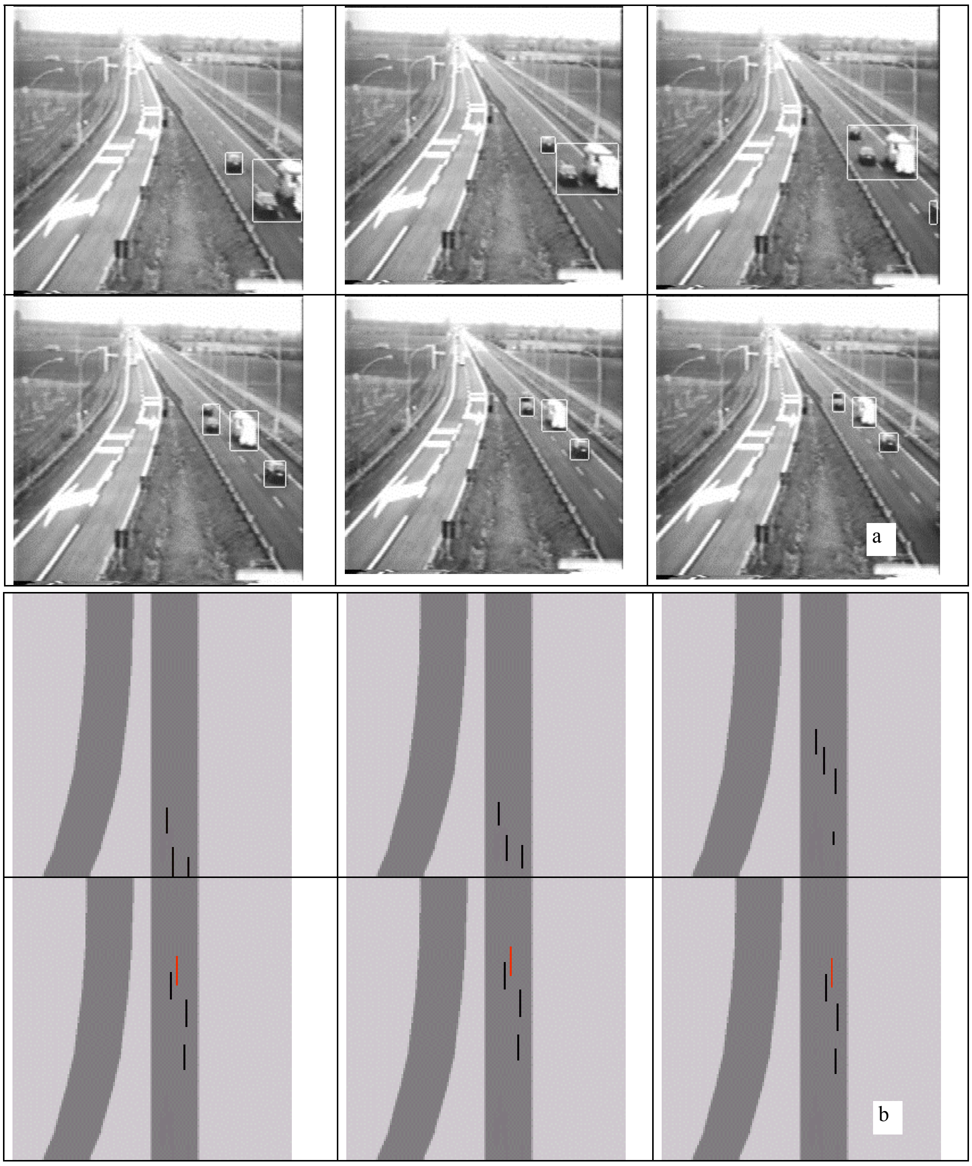

Fig. 6 – (a) A sequence of traffic where some occlusions occur, (b) display on the top view map of traffic parameters (position and speed of detected vehicles). Vehicle in ghost state are represented in red color.

7. Traffic flow analysis

Dangerous events on a monitored road or highway can be recognized by classifying vehicle behaviour, e.g., vehicle stopped on an emergency lane or in the middle of the road, vehicles moving with unusual trajectories, etc. An important feature for vehicle behaviour analysis is represented by the vehicle trajectory. Statistical models of the vehicle trajectories can be build up in order to mathematically describe the behaviour of each moving vehicle in the scene. This model should be defined a-priori and it depends on the characteristics of the monitored scene. A set of events that can be considered as potentially dangerous on a highway is the following: (a) vehicles that move along the highway lane with a non linear trajectory, (b) vehicles that decelerate and stop rapidly in the middle of the lane (i.e. an incident has occurred), (c) vehicles that exit out of the lane in the right side, (d) high number of vehicles (possible traffic congestion).

Figure 7 details the general architecture of an event recognition module [22]. The input of this module is represented by a set of observations taken by the tracking and classification modules over N consecutive frames of the sequence. In particular, the estimated object position on the ground plane, the related classification and the estimated object trajectory are considered. In particular, a trajectory analysis module computes from the set of N consecutive object positions an approximation of the trajectory by means of Bezier curves.

A winner-takes-all classification module uses the static object classification obtained on each of the previous N frames, to compute the final classification of the object moving in the whole sequence. The matching process necessary to associate the same object over consecutive frames is performed by the tracking module. Vehicle motion constraints, e.g., maximum vehicle speed, constant speed and direction, etc., can be applied to reduce the computational complexity of the matching process. The class which receives the maximum number of occurrences is selected as the class of the detected object. Finally, the event recognition module integrates information about the object class and the estimated trajectory to individuate suspicious events.

7.1. Trajectory analysis

The tracking module extracts from each image an estimated measure (x_i,y_i) of the position of the vehicle on the ground plane (Z=0). The trajectory analysis module processes a set of N consecutive vehicle positions on the road plane, $(x_{t-N},y_{t-N}),(x_{t-N+1},y_{t-N+1}),\ldots,(x_t,y_t)$, where t represents the current time instant. First, at each time t, the module computes the displacement D_t of the tracked vehicle,

$$D_t = \sqrt{(x_t - x_{t-N})^2 + (x_t - x_{t-N})^2} \tag{10}$$

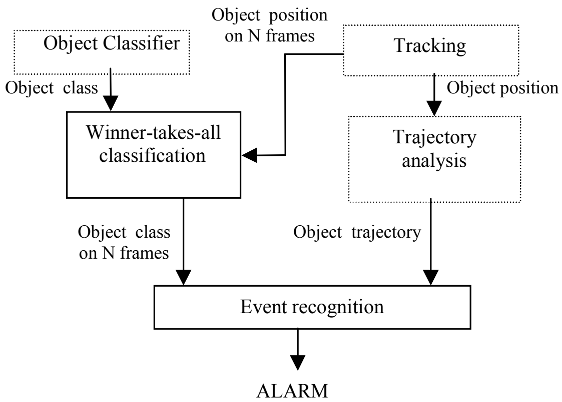

Fig. 7 – Architecture of the event detection module

If this displacement is lower than a fixed threshold Th_d, chosen by trials and according to the range of possible speeds that vehicles can reach in the considered application, the vehicle is considered in a stationary state, otherwise it is considered in a moving state. The stationary state comprehends two different situations: (a) the vehicle is stopped in a given position on the road or highway lane or (b) the vehicle is moving slowly in a restricted area of the monitored scene.

Then, the trajectory analysis module uses the Bezier curves to approximate the vehicle trajectory by starting from a set of of N consecutive object positions. Let $\mathbf{B}_0,\ldots,\mathbf{B}_N$ be $(N+1)$ points belonging to the road plane. The parametric Bezier curve [52] generated by the above $(N+1)$ points is given by

$$\mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{B}_i J_{n,i}(t) \qquad 0 \le t \le 1 \tag{11}$$

where $J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$.

Figure 8a and Figure 8b show some atypical and typical trajectories of vehicles on the 2D top view map. Atypical trajectories refer to vehicles that move several times from a lane to another, vehicle that come a stop or vehicle that for incidents exit from the lane.

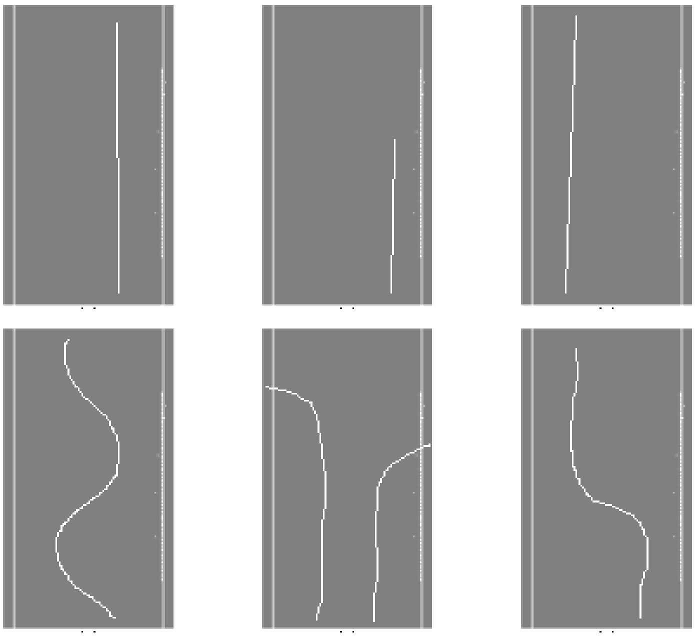

Fig. 8 (a,b) Normal and unusual trajectories of vehicles moving on a road.

7.2. Dangerous event detection

The event detection module produces alarm signals when abnormal trajectories are found. Table I shows the situations that the system can recognize as potentially dangerous and the level of warning signalled to the remote operator:

Tab. 1 – Set of possible alarm situations

Situation	*Signal*
Vehicle stopped on the emergency lane	Warning
Vehicle stopped on a forbidden area	Danger
Heavy traffic flow	Warning
Vehicle too fast	Warning
Dangerous trajectory	ALARM

A functional diagram of the event detection module is shown in Figure 9.

All information computed from each frame by the low-level system modules is provided to the event detection module. In particular, the position and the label of the blobs extracted at each frame of the sequence are used. Due to the different motion of the vehicles and to their different speeds, the order of blob extraction may change between two consecutive frames and occlusions can occur.

This problem is overcome by the tracking module which compares the new position with that estimated at the previous frame and executes the matching (also verifying that the distance has to be less than a fixed threshold). All the objects mismatched are labelled with a flag which indicates the ghost-status, and they are kept into the data structure until are matched again or until they reach a maximum number of consecutive ghost status (then they are erased). A slightly different procedure is adopted for new objects, which uses a priori knowledge. This procedure assumes that if one or more objects appear in a specific area of the monitored scene, they are instantiated with some defaults parameters (like the speed).

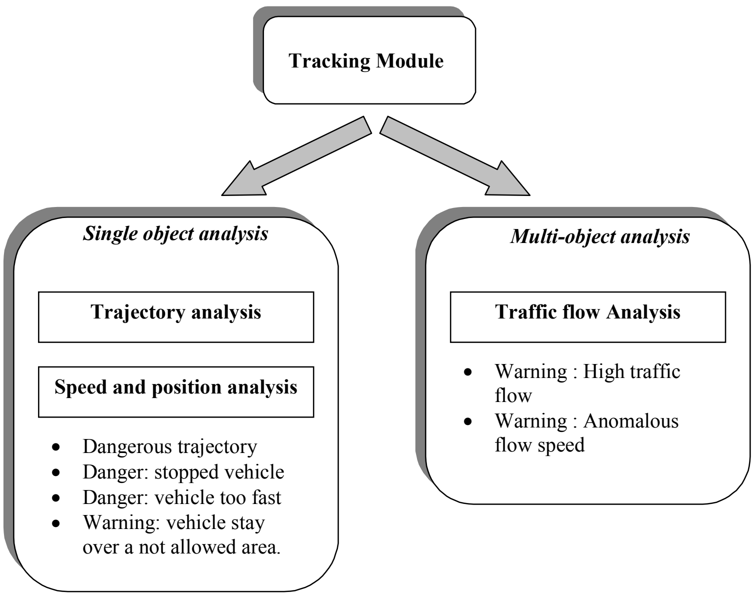

Fig. 9 – Functional diagram of the event detection module

The tracking stage is very important for the overall system behaviour, clearly if some matches are wrong then all the following assessments are influenced. However, experimental results showed that in a normal traffic conditions it almost never happens. On the other hand, when the traffic is heavy the number of objects is high and the distances between them are some times so small that mismatches may happen quite frequently. To cope with this problem a multiple object analysis (MOA) module has been introduced. It provides an average traffic flow assessment in order to recognize some particular events (e.g., high traffic flow) and regulates the level of warnings adopted by a single object analysis (SOA) module in each situation.

The SOA module executes some controls on each single vehicle. Firstly, it verifies the average speed of the vehicle. If the vehicle is stopped in the middle of the lane, then a signal of dangerous is activated (i.e. a car accident) and sent to the remote operator. A warning signal is activated if the vehicle stops on the emergency lane. On the opposite, a signal will advert the operator if a vehicle is running over the speed limits.

Secondly, the trajectory of the vehicle is analyzed by extracting the Bezier‘s curve which approximates the positions of the vehicle over the frames of the sequence and it is provided to the object recognition for classification.

The MOA module handles the traffic flow information: in particular it calculates the number of vehicles on the frame (trivial), the average speed of the flow and his variance. With these parameters an ANN can be trained to recognize various traffic flow situations, e.g., it is possible to detect the flow acceleration or deceleration which could be caused by a car accident.

8. Conclusions

An exhaustive survey of visual-based traffic monitoring systems and an overview of the principal computer vision techniques normally applied in these systems have been shown. In particular, the state of the art on traffic monitoring systems and their general architecture have been deeply investigated.

Furthermore, the principal modules of a visual surveillance systems have been described. The principal techniques applied to perform change detection, object recognition, and tracking have been shown.

Finally the traffic flow analysis for the detection of dangerous events on monitored road and highways has been considered.

The reported techniques represent the base for advanced visual-based traffic monitoring systems which aim to increase the safety in road transportation. This will be achieved by enhancing the traffic monitoring by helping human operators and by introducing driving assistance systems to support humans in keeping a safe distance from other vehicles, in detecting obstacles under all weather conditions etc.

The development and deployment of advanced monitoring systems will hopefully reduce the number of injuries and fatalities deriving from road accidents.

Appendix I

Let $\phi 1,..,\phi 7$, be moment invariants introduced by Hu, i.e.,

$$\Phi_1 = \mu_{2,0} + \mu_{0,2} \tag{A1}$$

$$\Phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \tag{A2}$$

$$\Phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \tag{A3}$$

$$\Phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2 \tag{A4}$$

$$\Phi_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2] + \\ + (3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] \tag{A5}$$

$$\Phi_6 = (\mu_{2,0} - \mu_{0,2})(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2 + 4\mu_{1,1}[(\mu_{3,0} + \mu_{1,2})(\mu_{2,1} + \mu_{0,3})] \tag{A6}$$

$$\Phi_7 = 3(\mu_{2,1} + \mu_{0,3})(\mu_{3,0} - \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2] - [3(\mu_{2,1} + \mu_{0,3})^2] - \\ - (\mu_{3,0} - 3\mu_{1,2})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] \tag{A7}$$

Acknowledgments

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Distributed systems for multisensor recognition with augmented perception for ambient security and customization" (2002-2004).

References

1. "Transport safety performance in the EU - a statistical overview", Report of the European Transport Safety Council, July 2003.
2. V. Anastassopoulos and A.N. Venetsanopoulos, "The classification properties of the Pecstrum and its use for Pattern Identification", *Circuits Systems and Signal Process*, vol. 10, no. 3, 1991, pp. 293-325.
3. Aoki, M. Masayoshi, "Imaging and analysis of traffic scene", *International Conference on Image Processing*, Kobe Japan, 1999, pp. 1-5.
4. Bertozzi M. and Broggi A., GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection, *IEEE Transaction on Image Processing*, Vol. 7, No. 1, January 1998, pp. 62-81.
5. Broggi, M. Bertozzi, A. Fascioli, C.G. Lo Bianco and A. Piazzi, "Visual Perception of Obstacles and Vehicles for Platooning", *IEEE Transaction on Intelligent Transportation Systems*, Vol. 1, No. 3, September 2000, pp. 164-176.
6. M. Betke, E. Haritaoglu and L.S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle", *Machine Vision and Applications*, Vol 12, No. 2, 2000, pp. 69-83.
7. L. Blincoe, A. Seay, E. Zaloshnja, T..Miller, E. Romano, S.Luchter, R.Spicer, "The Economic Impact of Motor Vehicle Crashes, 2000", Report DOT 809 446, May 2000, U.S. Department of Transportation, National Highway Traffic Safety Administration (NHTSA).
8. J. Boyd, J. Meloche and Y. Vardi, "Statistical Tracking in Video Traffic Surveillance", *In Proceeding of the IEEE International Conference on Computer Vision*, Corfu Greece, September 20-25 1999, pp. 163-168.
9. K.J. Bradshaw, I.D. Reid, and D.W. Murray, "The active recovery of 3D motion trajectories and their use in prediction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, 1997, pp. 219-234.
10. R. Chapuis, A. Potelle, J.L.Brame and F.Chausse, "Real-Time Vehicle Trajectory Supervision on the Highway", *International Journal on Robotics Research*, Vol. 14, No. 6, December 1995, pp. 531-542.
11. R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, andO. Hasegawa, "A System for Video Surveillance and Monitoring", Tech. Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May, 2000.
12. S. A. D Agostino, P. W. Shuldiner, "Application of video/machine vision technology in traffic data analysis", Proc. SPIE Vol. 2344, January 1995 , Intelligent Vehicle Highway Systems; Richard J. Becherer; Ed, p. 207-214.
13. E. Durucan and T. Ebrahimi, "Change detection and background extraction by linear algebra", *Proceedings of the IEEE*, Vol. 89, n°.10, pp.1368-1381, October 2001.

14. W. Enkelmann , "Video-Based Driver Assistance - From Basic Functions to Applications", *International Journal of Computer Vision*, Vol. 45, No. 3, pp 201-221, 2001.
15. N.J. Ferrier, S.M. Roew and A. Blake, "Realtime traffic monitoring", in *Proc. of 2nd IEEE Workshop on Application of Computer Vision*, Sarasota, Florida, December 1994, pp. 81-88.
16. J.M. Ferryman, S.J. Maybank, A.D. Worrall, "Visual Surveillance for Moving Vehicles", *International Journal on Computer Vision,* Vol. 37, No. 2, June 2000, pp. 187-197.
17. G.L. Foresti, "Detecting Multiple Objects Under Partial Occlusion by Integrating Classification and Tracking Approaches", *International Journal of Imaging Systems and Technology*, Vol. 11, 2001, pp. 263-276.
18. G.L. Foresti, "Object Detection and Tracking in Time-Varying and Badly Illuminated Outdoor Environments", *Optical Engineering*, Vol. 37, no. 9, 1998, pp. 2550-2564.
19. G.L. Foresti, "A Real-time System for Video Surveillance of Unattended Outdoor Environments", *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 8, No. 6, 1998, pp. 697-704.
20. G.L. Foresti, "Outdoor Scene Classification by a Neural Tree Based Approach", *Pattern Analysis and Applications*, Vol. 2, 1999, pp. 129-142.
21. G.L. Foresti, "Real-time detection of multiple moving objects in complex image sequences", *Int. Journal of Imaging Systems and Technology*, Vol. 10, 1999, pp. 305-317.
22. G.L. Foresti and F. Roli, "Real-time recognition of suspicious events for advanced visual-based surveillance" in *Multimedia Video-Based Surveillance Systems: from User Requirements to Research Solutions*, G.L. Foresti, C.S. Regazzoni and P. Mahonen (eds.), Kluwer Academic Publishers, 2000, pp. 84-93.
23. J. Goldbeck, B. Huertgen, S. Ernst, L. Kelch, " Lane following combining vision and DGPS", *Image and Vision Computing*, Vol. 18, No. 5, April 2000, pp. 425-433.
24. A.E. Grace, D. Pycock, H.T. Tillotson, and M.S. Snaith, "Active shape from stereo for highway inspection", *Machine Vision and Applications*, vol. 12, no. 1, 2000, pp. 7-15.
25. M. Haag, H.H. Nagel, "Incremental recognition of traffic situations from video image sequences", *Image and Vision Computing*, Vol. 18, No 2, January 2000, pp.137-153.
26. U. Handmann, T.Kalinke, C. Tzomakas, M. Werner and W.Von Seelen, "An image processing system for driver assistance", *Image and Vision Computing* , No. 5, April , pp. 367-376, 2000.
27. Haritaoglu, D. Harwood, and L.S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.
28. S. Haykin, Neural Networks: A Comprehensive Fundation, IEEE Press, MA, 1994.
29. J.E. Hollis D.J. Brown, I.C. Luckraft and C.R. Gent, "Feature vectors for road vehicle scene classification", *Neural Networks*, vol. 9, no. 2, 1996, pp. 337-344.
30. Houghton, G. Hobson, L. Seed, and R. Tozer, "Automatic monitoring of vehicles at road junctions," *Traffic Engineering Control*, vol. 28, n°.10, pp. 541-453, October, 1987.
31. M.K. Hu, "Visual pattern recognition by moment invariant", *IEEE Transaction on Information Theory*, vol. 8, 1962, pp. 179-187.
32. R. Inigo, "Application of machine vision to traffic monitoring and control," *IEEE Transactions on Vehicular Technology*, vol. 38, n°.3, pp. 112-122, August, 1989.
33. O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance", *In Proceedings of the 7th European Conference on Computer Vision*, 27 May- 2 June 2002, Copenhagen, Vol.4, pp. 343-357.
34. M-P. D. Jolly, S. Lakshmanan and A.K. Jain, "Segmentation and classification using deformable templates", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 293-308, 1996.
35. D.J. Kang, J.E. Ha and I.S. Kweon, "Fast object recognition using dynamic programming from combination of salient line groups", *Pattern Recognition*, Vol. 36, 2003, pp. 79-90.
36. M. Kilger, "A shadow handler in a video-based real-time traffic monitoring system", *IEEE Workshop on Applications of Computer Vision*, Palm Springs, CA, November 30-December 2, 1992, pp. 11-18.

37. X. Li, Z.Q. Liu and K.M. Leung, "Detection of vehicles from traffic scenes using fuzzy integrals", *Pattern Recognition*, Vol.35, No. 4, April 2002, pp. 967-980.
38. S. Loncaric, "A survey of shape analysis techniques", *Pattern Recognition*, Vol. 31, No. 8, 1998, pp. 983-1001.
39. Medonca P.R.S. and R. Cipolla, " A simple technique for self-calibration", *In proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Fort Collins, Colorado, Vol. I, June 1999, pp. 500-505
40. P. Maragos and R.W. Schafer, "Morphological skeleton representation and coding of binary images", *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 34, 1986, pp. 1228-1244.
41. P.Maragos, "Pattern spectrum and multi-shape representation", *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 11, 1989, pp. 701-716.
42. E. Marchand, P. Bouthemy, F. Chaumette, "A 2D-3D model-based approach to real-time visual tracking", *Image and Vision Computing*, Vol. 19, no. 13, 2001, pp. 941-955.
43. G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, Event Detection and Analysis from Video Streams, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, 2001, 873-889.
44. P. Michalopoulos, "Vehicle detection video through image processing: the AUTOSCOPE system", *IEEE Trans. on Vehicular Technology*, vol. 40, no.1, 1991, pp. 21-28.
45. Nair D. and Aggarwal J.K., Moving Obstacle Detection From A Navigating Robot, IEEE Transaction on Robotics and Automation, No 3, June 1998, pp. 404-416.
46. J.R. Parker, "Gray-level thresholding in badly illuminated images", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 813-819.
47. T. Pavlidis, "Algorithms for Shape Analysis of Contours and Waveforms", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, 1980, pp. 301-312.
48. B.T. Phong, "Illumination for computer generated pictures," *IEEE Transaction on Communication*, Vol. 18, 1976, pp. 311-317.
49. R.J. Prokop and A.P. Reeves, "A Survey of Moment-based techniques for unoccluded object representation and recognition", *Graphical Models and Image Processing*, vol. 54, no. 5, 1992, pp. 438-460.
50. J.R. Quinlan, "Induction of decision trees", *Machine Learning*, vol. 1, 1986; pp. 81-106.
51. R. Jain, D. Militzer, and H.H. Nagel, "Separating non-stationary from stationary scene components in a sequence of real-world TV images," in *Proc. of Int. Joint Conference on Artificial Intelligence*, 1977, pp. 612-618.
52. D.F. Rogers and J.A. Adams, "Mathematical elements for computer graphics", McGraw Hill, 1990.
53. P.L. Rosin, "Thresholding for Change Detection", *Computer Vision and Image Understanding*, Vol.86, n°.2, pp. 79-95, May 2002.
54. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.
55. I.K. Sethi, "Entropy nets: from decision trees to neural networks", *Proceedings of the IEEE*, vol. 78, no. 10, 1990, pp. 1605-1613.
56. C.E.Smith,S.A.Brandt,C.A.Richards and N.P.Papanikolopoulos. Visual tracking for intelligent vehicle-highway systems. *IEEE Transaction on Vehicular Technology*, Vol 46, No.4, 1996, pp. 732-743.
57. H. Schneiderman and M. Nashman, "A Discriminating Feature Tracker for Vision-Based Autonomous Driving", *IEEE Transaction on Robotics and Automation*, Vol. , No 10, 1999, pp. 769-775.
58. K. Skifstad and A. Jain, "Illumination independent change detection for real world sequence", *Computer Vision, Graphics, and Image Processing*, Vol. 46, 1989, pp 387-399.
59. L. Snidaro and G.L. Foresti, "Real-time thresholding with Euler numbers", *Pattern Recognition Letters*, Vol. 24, n. 9-10, pp. 1533-1544, June 2003.
60. C. Stauffer and E. Grimson, "Learning Patterns of Activity Using Real-Time Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22(8), pp. pp. 747-757, August 2000.
61. C. Stiller, J. Hipp, C. Rössig, A. Ewald, "Multisensor obstacle detection and tracking", *Image and Vision Computing, No. 5, April 2000, pp. 389-396.*

62. G.D Sullivan, "Visual interpretation of known objects in constrained scenes", *Phil. Trans. Royal Society of London B.*, vol. 337, March 25-26 1992, pp. 109-118.
63. H. Taniguchi, T. Nakamura and H. Furusawa, "Methods of Traffic Flow Measurement using Spatio-Temporal Image", *In Proceeding of International Conference on Image Processing*, Kobe, Japan, Oct. 25-28, 1999, 28AS1.
64. R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE Journal of Robotics and Automation*, Vol. RA-3, n. 4, pp. 323-344, 1987.
65. J. Wood, "Invariant pattern recognition: a review", *Pattern Recognition*, vol. 29, no. 10, 1996, pp. 1-17.
66. K. Yoon, D. DeMenthon, and D. Doermann, "Event Detection from MPEG Video in the Compressed Domain", in *Proc. of 15th International Conference on Pattern Recognition*, Barcelona, Spain, Sep.3-7, 2000, pp. 819-822.
67. X. Yuan, Y-J. Lu and S. Sarraf, "Computer vision system for automatic vehicle classification", *Journal of Transportation Engineering*, vol. 120, no. 6, pp. 861-876, 1994.
68. Zhigang Zhu, Guangyou Xu, Bo Yang, Dingji Shi, and Xueyin Lin, "VISATRAM: a real-time vision system for automatic traffic monitoring", *Image and Vision Computing*, vol. 18, no. 10, 2000, pp. 781-794.
69. T. Zielke, M. Brauckmann and W. Von Seelen, "Intensity and edge-based symmetry detection with an application to car following", *Computer Vision, Graphics and Image Processing: Image Understanding,* Vol. 58, no. 2, 1993, pp. 177-190.
70. F. Ziliani and A. Cavallaro, "Image analysis for video surveillance based on spatial regularization of a statistical model-based change detection", *Real-Time Imaging*, Vol. 7, pp.389-399, 2001
71. Y. Bar-Shalom and W. D. Blair (editors), "Multitarget multisensor tracking: applications and advances Volume III", Artech House, 2000.
72. Dorin Comanesciu, V. Ramesh and Peter Meer, "Real-time Tracking of Non-Rigid Objects using Mean Shift," *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, 2000.