Processing web-cam images by a neuro-fuzzy approach for vehicular traffic monitoring

A. Faro, D. Giordano, and C. Spampinato

Abstract**—** Traffic management in an urban area is highly facilitated by the knowledge of the traffic conditions in every street or highway involved in the vehicular mobility system. Aim of the paper is to propose a neuro-fuzzy approach able to compute the main parameters of a traffic system, i.e., car density, velocity and flow, by using the images collected by the web-cams located at the crossroads of the traffic network. The performances of this approach encourage its application when the traffic system is far from the saturation. A fuzzy model is also outlined to evaluate when it is suitable to use more accurate, even if more time consuming, algorithms for measuring traffic conditions near to saturation.

Keywords**—** Neuro-fuzzy networks, Computer vision, Fuzzy systems, Intelligent Transportation System.

I. Introduction

AN accurate monitoring of the vehicular traffic flows is at the basis of an effective traffic management for either highways control or optimization of the urban traffic lights cycles. As is known, the traffic flows may be measured by cables that count the number of vehicles passing through a given section in the time unit. Density and velocity measurements, directly not detectable from cables, could be deduced from a mathematical model, e.g. by the flow-density relation characterizing the streets [1]. Using images for traffic monitoring is more suitable since all the traffic parameters can be measured experimentally by images on traffic collected by cameras without deriving car density and velocity from traffic flows by means of approximate mathematical formulas. Moreover more or less all the relevant traffic conditions have a visual nature as for example detecting accidents, congestion, or maintenance work on the streets. This justifies the technical convenience of using cameras instead of cables. However, the use of a high number of cameras and the necessity of sending high volume of data to the processing nodes for computing the optimal values of the traffic light cycles may be too expensive. Thus studies and practical proposals on how to use economical cameras, such as web cams, for traffic monitoring and adopting a distributed processing to reduce the volume of data to be transferred over the information network are more and more needed.

In [2] the authors have proposed two algorithms, called W4-like and H algorithms, to compute the traffic parameters starting from the images taken by web-cams. Aim of the paper is to investigate if and when a neuro-fuzzy approach to traffic monitoring from web-cam images is better that the mentioned two methods. A fuzzy model is also presented to decide in real time the algorithm most suitable for the current traffic condition with the aim of avoiding to process the images when the traffic changes slowly or to apply accurate algorithms when the traffic is far from the saturation.

Sect.2 briefly recalls the W4-like and H algorithms by providing also a comparison of the performances attainable by such algorithms. Sect.3 discusses how a Neuro-Fuzzy (NF) approach may be used for traffic measurements. A comparison among the W4-like, H and NF algorithms is also provided in this section. Sect.4 presents the fuzzy model to decide what algorithm has to be used depending on the current traffic conditions.

II. Processing web-cam images by W4 and H algorithms

This section briefly recalls two algorithms introduced by the authors to evaluate the traffic conditions starting from the images taken by web-cams. A comparison at the end of the section will clarify when using one algorithm rather than the other.

A. W4-like algorithm

In this algorithm, inspired by the W4 algorithm [3], each pixel p(x, y) is characterized by three parameters computed by processing some images taken from the web-cam during an initial learning phase, i.e.,: the maximum intensity detected M(x, y), the minimum intensity detected N(x, y), and the maximum difference D(x, y) between the intensities of the same pixel p(x, y) in two images belonging to the learning set. A pixel, whose current intensity is I(x, y), is considered belonging to the foreground if it satisfies the following condition:

$$\mathrm{Abs}[M(x, y) - I(x, y)] > D(x, y) \text{ and}$$
$$\mathrm{Abs}[N(x, y) - I(x, y)] > D(x, y)$$

A.Faro is with Dipartimento di Ingegneria Informatica e Telecomunicazioni, Universita' di Catania, viale Andrea Doria 6, 95125, Italy (phone: 39 0957382372; fax: 39 095 338280; e-mail: , afaro@diit.unict.it).
D.Giordano is with Dipartimento di Ingegneria Informatica e Telecomunicazioni, Universita' di Catania, viale Andrea Doria 6, 95125, Italy (phone: 39 0957382371; fax: 39 095 338280; e-mail: , dgiordan@diit.unict.it).
C.Spampinato is with Dipartimento di Ingegneria Informatica e Telecomunicazioni, Universita' di Catania, viale Andrea Doria 6, 95125, Italy (phone: 39 0957382372; fax: 39 095 338280; e-mail: , cspampin@diit.unict.it).

The density of cars is then obtained by the ratio between the number of pixels belonging to the foreground divided by the overall number of pixels in the scene.

B. *H-algorithm*

This algorithm determines if a pixel (x, y) belongs to the background or to the foreground by using a more accurate statistics about the possible values of the pixels. The result of the H-algorithm is a frame characterized by a binary function P(x, y) such that if P(x, y) = 1, then the pixel p(x, y) belongs to the foreground else it belongs to the background. Any algorithm belonging to the "selected component labeling" family [4] may be used to classify the pixels into N classes, being N the number of objects present on the scene.

C. *Comparison between W4-like and H-algorithms*

Although the H-algorithm is fast, the W4-like algorithm is about 30% faster, even if 15% less precise than the H-algorithm. This confirms the effectiveness of our choice of using the W4-like algorithm without executing the "selected component" algorithm to evaluate, depending on the current value of the car density δ, if the more accurate H-algorithm, followed by the "selected component" algorithm, has to be executed to compute the car density δ(t) and consequently to update the values of the car flows.

III. Processing web-cam images by a Neuro-Fuzzy net

Both the previous algorithms compute the traffic parameters by extracting the information belonging to the foreground with respect to the one belonging to the background. Experimentally we have found that the weather conditions influence the performances of the W4-like algorithm, whereas the change of such conditions do no influence the H4 algorithm. Thus a fast algorithm for web-cam image processing that does not depend on the environment luminosity is useful to replace the W4 algorithm by one characterized by lower processing time and greater accuracy.

With this aim in mind we present in this section how to monitor the car traffic by a neuro-fuzzy approach since a neuro-fuzzy model is able to approximate non-linear processes such as the one of detecting the presence of objects (i.e., cars) having different geometric features moving on a background that may change due to the external luminosity. Executing for a sequence of frames the cross-correlation between the variables that may affect the car detection and how much a car is present in such frames we have found that three subsequent frames are enough for identifying a model for car traffic detection and that the main variables to take into account are: the average value of the pixels intensity (i.e., μ(t-2), μ (t-1), μ (t)), the maximum and the minimum pixel intensities (Max(t) and Min(t)), the cross-correlation between two subsequent frames (i.e., Max(Corr(t-2,t-1)) and Max(Corr(t-1,t))) and the variance (i.e., σ^2(t-2), σ^2 (t-1), σ^2 (t)). Let us note that all the mentioned elaborations are carried out with respect to a virtual rectangular spire contained in the frames. Such spire should be small enough to avoid that two cars are present contemporaneously in it. Of course under a certain spire width, the presence of a car cannot be distinguished from the noise in background. As an example, fig.1 shows the image belonging to the spire in three subsequent frames taken by a web-cam. Starting from images of this kind, it is possible to compute the ten inputs (see fig.2) to be given at every instant t to the neuro-fuzzy network.

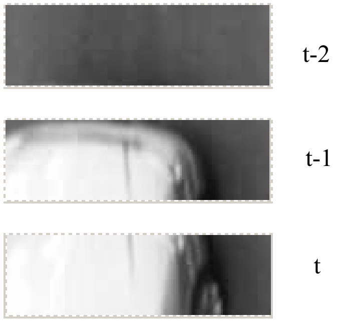

Fig.1 - Three configurations of the virtual spire: a) no car inside the spire at time t-2, b) a car is entering into the spire at time t-1 , and c) a car is traversing the spire at time t.

Input1	Input 2	Input3	Input4	Input5
μ(t-2)	μ (t-1)	μ (t)	Max(t)	Min(t)

Input6	Input7
Max(Corr(t-2,t-1))	Max(Corr(t-1,t))

Input8	Input9	Input10
σ^2(t-2)	σ^2 (t-1)	σ^2 (t)

Fig.2 – The ten inputs of the neuro-fuzzy network

Fig.3 shows the structure of the neuro-fuzzy network adopted to discover if the spire contains or not a car at a certain instant t. For simplicity the structure deals with two inputs instead of the ten inputs we have passed to the neuro-fuzzy network in our experiments. The blocks labeled by $MF1_i$ and $MF2_i$ respectively represent the membership functions related to input $_i$,i.e., $input_i$ is low, $input_i$ is high. The block Π_{rs} provides the value v_{rs} of the rule antecedents such as "$input_r$ is low and $input_s$ is high". In our case v_{rs} is given by the product of the fuzzy values associated to the basic components, i.e., "$input_r$ is low" and "$input_s$ is high". The outputs of Π_{rs} are multiplied by weights w and are passed to a summation block to produce the output of the network. The difference between the real output and the one computed by the net is used to change the values of the weights w and of the form of the memberships functions.

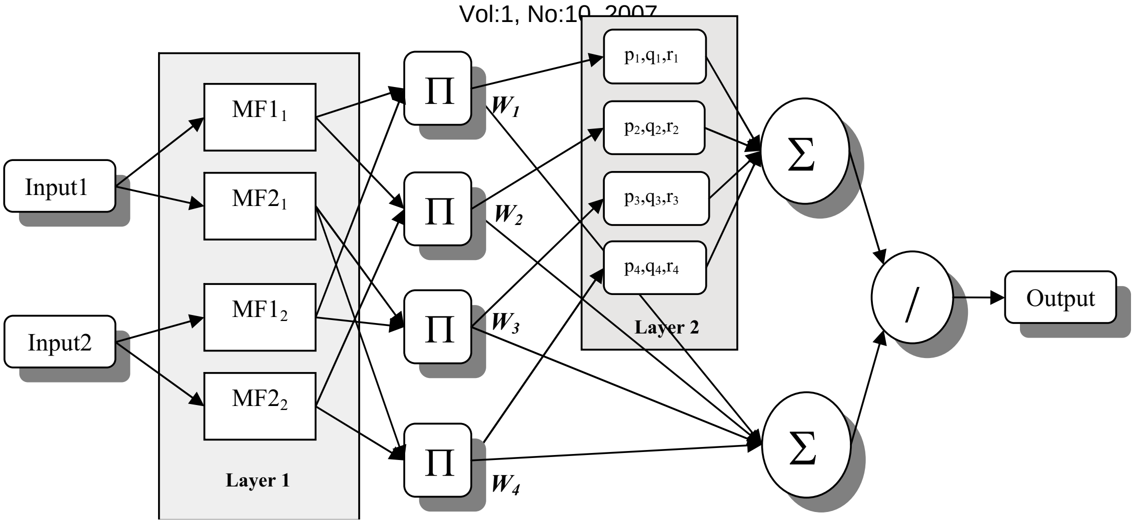

Fig.3 – Neuro-fuzzy network adopted for processing the images taken by the web-cams

This way of proceeding produces the net will converge to the final configuration according to the Mamdani algorithm [5]. However, to increase precision and to speed up the convergence, we have adopted the Sugeno approach [5]. For this reason the outputs of the block Π_{rs} are passed also to other blocks contained in a second layer whose outputs are given by the formulas such as:

p F("input$_r$ is low", input$_r$) + q F("input$_s$ is high", input$_s$) + r

According to the Sugeno algorithm, also known as ANFIS (i.e., adaptive neuro-fuzzy inference system), the value of p, q and r will change if the output obtained by using the mentioned Mamdani algorithm is better than the output of the second layer, otherwise p, q and r will not be changed and both the weights w and the form of the mentioned membership functions will be changed. The percentage of success in computing the number of cars traversing the spire in the time unit of the adopted NF algorithm is about 92% in all the weather conditions. The NF algorithm is 10% more accurate of the W4-like algorithm and has more or less the same time performance of the W4-like algorithm. The H algorithm is more time consuming (about 30% more than the previous algorithms), but it is preferable to obtain very accurate measurements since it reaches about 97% of success in pattern recognition.

Thus the best solution should be to execute locally, i.e., on the board associated to the web-cam, the evaluation of the traffic parameters by using both W4-like and NF algorithms. This will allows us to obtain experimentally all the traffic parameters: density δ from W4, flow λ from NF and average velocity $v = \lambda / \delta$. The web-cam images will be sent to a powerful processing node only when we are near to the density saturation and/or the flow is changing. suddenly according to the fuzzy rules explained in the next section.

IV. FUZZY RULES TO DECIDE THE ALGORITHM TO APPLY DEPENDING ON THE TRAFFIC CONDITIONS

As above said, it is convenient that both the W4-like and the NF algorithm are executed contemporaneously on the crossroad PC board to avoid sending images from the web cams to the processing nodes when the traffic situation is more or less the same of the one measured in the previous processing cycle. This evaluation may be performed in two main ways: by using a crisp formula such as

$$\mathrm{Abs}[\Delta\delta_s\,(t) - \Delta\delta_s\,(t_p)] < \tau$$

being τ a suitable threshold, or according to some fuzzy logic based rules. Following the former method implies to define many thresholds τ in a discretional way. As an example, three thresholds τ_L, τ_M and τ_H should be defined depending on if the current value of δ is low, medium or high. Interestingly the fuzzy logic approach is less discretional than the crisp one once the memberships of the adopted fuzzy functions are defined, and allows us to implement strategies of traffic regulation directly derived from common sense rules. This latter approach is then chosen. The functions adopted for limiting the data exchange between the web-cams and the processing nodes are: a) the fuzzy functions related to δ_{low} δ_{medium} and δ_{high} (fig.4a), and b) the densities at three subsequent times t_0, $t_1 = t_0 + \Delta t$ and $t_2 = t_0 + 2\,\Delta t$ (fig.4b).

On the basis of these fuzzy functions, the program running on the web-cam board is capable of evaluating by a fuzzy model the following conditions that do not require the image processing: a) if δ is low and the car flow changes slowly, and b) if δ is medium-high and the car flow changes very little.

In other words, if one of these rules holds, no further processing is carried out in this processing cycle and the traffic flows are assumed to be equal to the ones evaluated at the previous cycle, otherwise the image is processed locally or sent to a processing server for measuring the current car flow by the H-algorithm depending on if the car density is far from or near to the saturation point. To show how the model works, we assume that the fuzzy functions “essentially”, “most” and “little” are the ones in fig.5a.

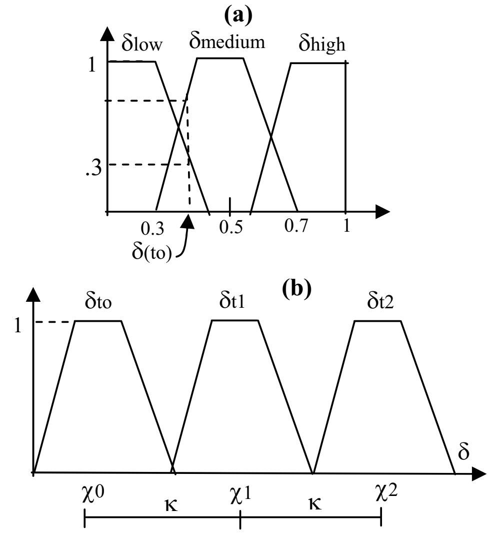

Fig.4 - Fuzzy functions representing δ (fig.4a) and the densities at three subsequent instants (fig.4b). The distance κ between the centers χ of two subsequent functions is the same, i.e., $\chi_{t1} - \chi_{t0} = \chi_{t2} - \chi_{t1} = \kappa$, being κ determined by the increment of the traffic flow Δλ measured at the time t_0 as follows: $\kappa = \Delta\lambda \ \Delta t$.

Then the fuzzy function F = “the car flow changes slowly” is formulated as follows:

at instant t_0 F_0 = Most δto + Little δt1,
at instant t_1 F_1 = Most δt1 + Little δt2.

Verifying that F is valid at both t_0 and t_1 means that:

δ(to) belongs to E_0 = “essentially” F_0, and
δ(t1) belongs to E_1 = “essentially” F_1.

As an example, from the forms of F_0 and E_0 shown in fig.5b, the agents may deduce that $\delta_o < \delta(to) < \delta_1$. Proceeding in a similar way, they may found the constraints of the density at instant t1 as follows: $\delta_1 < \delta(t1) < \delta_2$.

Thus when δ is low, if $\delta_o < \delta(to) < \delta_1$ and $\delta_1 < \delta(t1) < \delta_2$ no further processing is needed. The conditions to be verified to avoid further processing if ρ is medium or high may be found in a similar manner. On the contrary, when δ is a linear combination of low, medium and high, as for example δ = 0.3 low and 0.7 medium (see fig.4a), then both condition (a) and (b) should be applied and the values δ_o‘, δ_1‘ and δ_2‘ may be found as follows:

$$\delta_o‘ = 0.3\ \delta_o + 0.7\ \delta_1,\ \delta_1‘ = 0.3\ \delta_1 + 0.7\ \delta_2,\ \delta_2‘ = 0.3\ \delta_2 + 0.7\ \delta_3$$

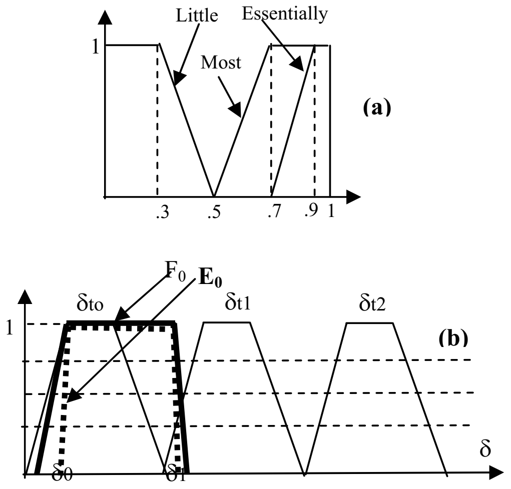

Fig.5 – Fuzzy functions used to limit the data exchange

V. Conclusion

The coordinated application of the W4-like, NF and H algorithms depending on suitable fuzzy rules dealing with traffic conditions allows us to implement a distributed traffic monitoring system whose nodes, located at the crossroads of the urban network, consist of a web-cam locally controlled by a simple computing board CB. This architecture has high performances and low costs. Providing CB with a fuzzy processor [6] the decision of using the NF algorithm in conjunction with the W4-like algorithm implemented on the CB or with the more accurate H algorithm implemented in some processing node of the monitoring system would be very fast. A prototype of such monitoring system whose nodes exchange data by wi-fi channels has been developed in cooperation with the Catania Municipality. It is under test for controlling the traffic of the main streets of the urban area.

References

[1] Sussman J.: Introduction to transportation systems - Artech House, 2000

[2] Faro A., Giordano D., Spampinato C.: Soft-computing agents processing web-cam images to optimize metropolitan traffic systems – Int. Conf. on Computer Vision and Graphics (ICCVG04), Warsaw, 2004

[3] Davis L. et alii: W4 real time surveillance of people and their activities - IEEE Trans on Pattern analysis and machine intelligence, v.22, N.8, 2000

[4] Fisher R. et alii, Connected Component labeling - http//homepages.inf. ed.ac.uk/ rbf/HIPR2/label.htm#1, 2003

[5] Byung-Doo Lee et alii: The application of the fuzzy reasoning to the opening games of 19x19Go – CITR-TR-15 2003- www.citr.auckland.ac.nz

[6] ST Microelectronics: The fuzzy processor ST52F13 – Catania 2002