Dynamic traffic light controller using machine vision and optimization algorithms

Fabio Espinosa, Camilo Gordillo, Robinson Jiménez, Oscar Avilés
GAV-Universidad Militar Nueva granada
{u1801067, u1801074, robinson.jimenez, oscar.aviles}@unimilitar.edu.co
Bogotá, Colombia

***Abstract*—This paper presents a fuzzy traffic controller that in an autonomous, centralized and optimal way, manages traffic flow in a group of intersections. The system obtains information from a network of cameras and through machine vision algorithms can detect the number of vehicles in each of the roads. Using this information, the fuzzy system selects the sequence of phases that optimize traffic flow globally. To evaluate the performance of the controller, a scenario was developed where it was possible to simulate through artificially created videos two adjacent intersections. System performance was compared versus fixed time controllers as they are currently the most used in the city of Bogota. As a control variable it was used the average waiting time of each vehicle. The results show that the system performance increases by about 20% over situations with heavy traffic conditions and that the controller is able to adapt smoothly to different flow changes.**

***Index Terms*—traffic control; computer vision; optimization; fuzzy control; object detection; classifiers.**

I. Introduction

NOWADAYS Bogota city presents a serious mobility problem, which affects a great part of the citizens and harms drastically its productivity and competitiveness [4]. One of the main reasons, which contributes to this situation, is the use of inefficient and obsolete traffic controllers, which are not capable to manage in an efficient way the traffic flow in the roads of the city. These fixed time controllers, require a periodical configuration based on statistical flow analyses, which generally do not reflect in an accurate way the real traffic flow conditions [1].

In order to solve this problem, new control techniques have been developed, allowing the creation of completely autonomous systems, that based in the data collected by a set of sensors (inductive, capacitive, acoustical), are able to manage in an optimal and dynamical way the vehicular flow [8].

Although the performance of these systems easily exceeds the performance of fixed time controllers, they present a maintenance problem mainly concerning the kind of sensors used. The great majority of the current solutions use the information provided by inductive sensors, which are installed directly into the asphalt. This kind of deployment leaves them exposed to all kind of physical interactions, which reduce drastically their useful life.

To avoid this problem, in this work a completely autonomous dynamical controller was developed, which is capable of manage in a coordinated and centralized way, the state of the traffic lights in a simulated scenario using the information provided by a set of cameras. This kind of sensor gives the system great installation flexibility, due to the possibility of strategic location within the control zone, avoiding the problems described above and increasing the durability, efficiency and profitability of the system.

The main contribution within the development of this controller is the use of a vehicular detection algorithm, which allows it to identify in an accurate way the number of vehicles present in each road. Besides, the controller has a diffuse optimization algorithm, which using the data provided by the detection algorithm switches the state of the traffic lights, ensuring a continuous, homogeneous, and fair traffic flow.

In the next section, the basic theory of vehicular traffic controllers and object detection techniques are summarized. At Section III the details of the system developed are presented. Section IV presents the characteristics of the test scenario designed while the results are showed at Section V. Finally at Section VI the conclusions achieved are presented.

II. Background and previous research

Bellow, the most relevant terms and investigations in the area of traffic controllers and vehicle detection inside images are mentioned.

A. *Traffic controllers*

There are two main kinds of traffic controllers: static ones and dynamical ones. The first ones are those where a sequence of actions previously programed are followed, while the other kind makes use of a certain acquisition method, which allows the system to identify the state of the traffic flow on the roads and guide his actions to optimize the traffic flow [2].

On the other hand, it is important to define a basic terminology: *phase*, *cycle* and *coordination*. A *phase* is a traffic signal which allows a flow of non-conflictive movements. For example, Phase 1 showed in Figure 1 allows traffic flow from west to east and vice versa. In the same way, a succession of phases which is repeated continuously is considered a *cycle*. Figure 1 shows a cycle made-up of 4 phases. Finally, *coordination* is the action of programing the signalized intersections in such a way, that the flow of a corridor can achieve a constant speed without detentions, generating what is known as *green waves*.

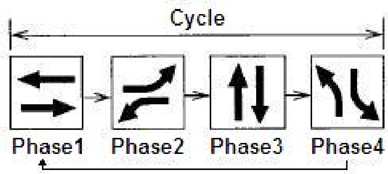


Figure 1. Four-phase cycle

Taking this into account, the action of controlling an intersection implies the determination of the phases which will be part of the cycle and also the duration of each of these phases.

One of the control strategies, that better results has achieved, is fuzzy logic. This technique allows controlling the traffic in an intersection in a similar way to the actions taken by a traffic officer, which turns out to be the main advantage of this kind of controllers. Lee et al. [3] proposed a distributed system capable of controlling the phase duration and the sequence of these, adapting to the different traffic situations, achieving a complete control over an intersections set.

B. Object detection

In the object detection field, there are two main strategies concerning the vehicle detection task: the first one is based on background and optical flow estimation, while the second one uses machine learning techniques. Background estimation analyzes the difference between a predefined model (image) of an empty road and an image of the incoming traffic, obtaining perturbations, that overlapped to the predefined model are interpreted as vehicles [16], [17]. A great portion of the investigations about machine learning methods has been framed to the '*on-road*' vehicle detection (a camera installed inside a car), instead of applications for traffic control on intersections. Examples of methods used within this area are: Boosted Cascade of haar Features, Sift (Scale Invariant Feature Transform) matching and neural networks.

In the same way, there is certain terminology which is important and will help understand this portion of the work. *Classifier* is an operator which uses the features of a data set, identifying the class or group to which each of these data belongs. *Boosting* is a meta-algorithm, which pretends to create a strong classifier through the addition of weak classifiers, and a *feature* is considered as an important piece of information [9].

There are plenty of investigations in the area of vehicle detection through images; the following are some of the most important researches in this field: in [10] and [11], an *on road* vehicle detector was developed through a Haar like feature detector, obtaining an accuracy detection of 88,6% and 76% respectively. In [12] and [13] authors used the background estimation technique with an efficiency rate over 90% in both cases. On the other hand, in [14] a morphological edge detector (SMED) was developed, which presents more insensitiveness to illumination changes than the background estimation, obtaining an accuracy of 95%.

III. Traffic control system

Figure 2 shows a physical diagram of the solution. At each intersection a computer is placed, this is in charge of acquiring images from a network of cameras. This computer is connected to a centralized server that processes the information and executes the detection and control algorithms. Finally all the decisions taken are sent back to each computer, which change the traffic lights depending on these orders.

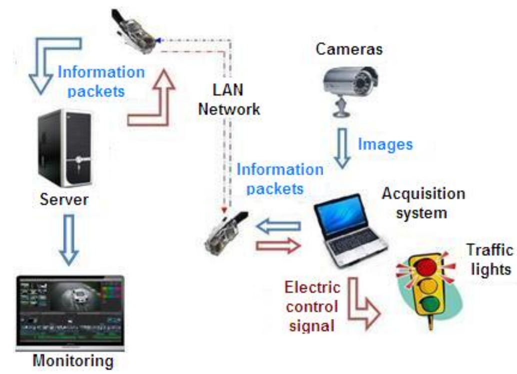


Figure 2. Physical diagram of the solution

A. Fuzzy control system

The controller developed is based on the model presented by Lee et al. in [3], which evaluates not only the variables related to the controlled intersection, but also analyzes the variables related to traffic flow at nearby intersections. This allows the system to operate in a coordinated way, thus generating so-called "*green waves*", avoiding unnecessary detentions for vehicles traveling through the roads and avoiding sending vehicles to areas of high congestion.

The controller basically consists of the three modules showed in Figure 3. The '*Next Phase*' Module is responsible for assessing the level of urgency of each of the phases that are not active, the *'Observation'* Module is in charge of studying traffic flow corresponding to the green phase, and the '*Decision*' Module determines whether the active phase at the intersection is changed to the one with the highest degree of urgency (depending on the module '*Next Phase*') or remains constant for a longer period.

It should be noted, that the level of urgency is just an analysis of how timely and favorable would be the exchange of the active phase.

The operating mode of each of these modules is described below:

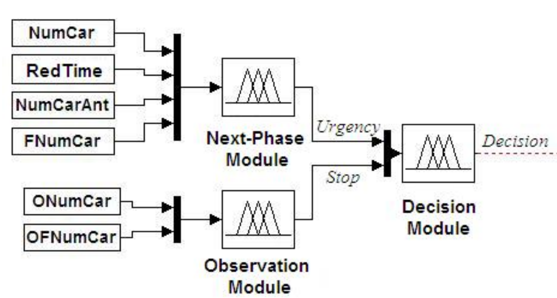


Figure 3. Schematic diagram of the controller

- *Next Phase Module*
  This is responsible for selecting among all the phases that are not active, the one whose level of urgency is higher. To achieve this, this module evaluates the urgency of each of the flows associated with each phase and the average of these values will be the level of urgency of the phase analyzed. For example, the level of urgency of the phase showed in Figure 4 is the average of the values obtained evaluating the north-south flow (green) and north-east flow (red).

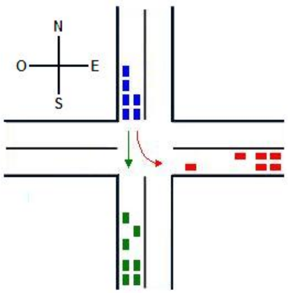


Figure 4. Sample situation

  To obtain the level of urgency of each flow, four variables are evaluated: *NumCar* is the number of vehicles waiting for the green signal, in Figure 4 they are represented in color blue; *RedTime* represents the number of periods that the evaluated phase has been deactivated; *NumCarAnt* is an estimate of the number of vehicles that could arrive from the lanes before the intersection, and *FNumCar* is the number of vehicles on the road in front of the intersection, for the north-south flow in Figure 4 this variable is represented with green color. In this way, the variables *RedTime* and *NumCar* reflect traffic conditions locally, while *NumCarAnt* and *FNumCar* allow the system to coordinate different neighboring intersections.
  Figure 5 shows the Fuzzy Set of this module and Table I presents some of its rules. For example, R2 states that if the number of vehicles waiting to cross is High (*NumCar* = H), the number of periods in which the analyzed phase has not been active is High (*RedTime* = H) and the number of vehicles waiting in the following lane is Low (*FNumCar* = L), then the urgency of this phase will be very high (*Urgency* = VH).

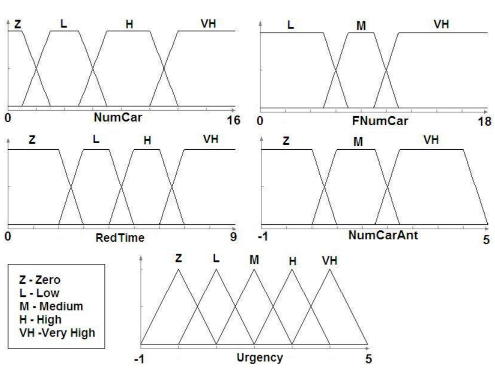


Figure 5. The Fuzzy Set of *NumCar*, *FNumCar*, *RedTime*, *NumCarAnt* and *Urgency*

Table I
SOME RULES OF THE NEXT PHASE MODULE

	NumCar	RedTime	NumCarAnt	FNumCar	Urgency
R1	Z		Z	L	L
R2	H	H		L	VH
R3	H	L		VH	L
R4	M	L	VH		H
...	...	...	...	...	...

- *Observation Module*
  This module is responsible for assessing traffic conditions for the active phase and determines, how timely it would be to stop that phase. The fuzzy rules of this module have two inputs and one output: *ONumCar* indicates the number of cars that still are on stanby; *OFNumCar* represents the number of vehicles at the next intersection and *Stop* is the output of the module and indicates, whether or not should be necessary to stop the phase. The behavior of the input variables is very similar to variables *NumCar* and *FNumCar*, therefore their fuzzy sets are equal. Figure 6 shows the Fuzzy Set for the *Stop* variable.
  Table II presents some rules of this module. R4 indicates that if the number of vehicles waiting for the active phase is still high (*ONumCar* = H) and the number of vehicles in the following lane is high too (*OFNumCar* = H), then the phase must be stopped (*Stop* = Yes). This is because it would be a waste of time to allow a flow that will be obstructed later.
- *Decision Module*
  This module decides whether or not change the active phase. The inputs in this module are *Urgency* and *Stop* and the output is *Decision*. The two input variables are the outputs of the modules '*Next Phase*' and '*Observation*', respectively. The module changes the active phase for that which is candidate, as long as the result of the defuzzification is above a given threshold.
  Table III shows some of the rules of this module. The first rule indicates that, although the candidate phase has a medium congestion (*Urgency* = M), if the Stop level of the active phase is low (*Stop* = N), then the module will

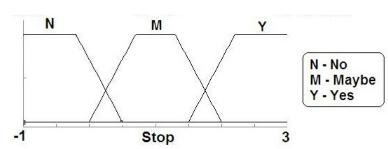


Figure 6. The Fuzzy Set of *Stop*

Table II
SOME RULES OF THE OBSERVATION MODULE

	ONumCar	OFNumCar	Stop
R1	Z		Y
R2	H	Z	N
R3	L	L	M
R4	H	H	Y
...	...	...	...

Table III
SOME RULES OF THE DECISION MODULE

	Urgency	Stop	Decision
R1	M	N	N
R2	H	M	Y
R3	VH	N	Y
R4	Z	Y	N
...	...	...	...

have to keep the same phase (*Decision* = N, no change). The Fuzzy Set of this module is presented in Figure 7 (*Urgency* and *Stop* variables appear in the previous modules).

B. Detection algorithms

As a detection algorithm, the one proposed in [5], [6] is used, this consist of a Haar features classifier cascade; accordingly to several authors [9], [10], [15] this method presents better or at least similar performance, than the best previous object detector systems. The implementation of this method is made up of two big phases, one dedicated to the training of the classifiers through a machine learning algorithm called Adaboost and the construction of the cascade, and the other where the detection is adapted to the own needs of the interest object and the context where these objects exist.

Within training phase, Adaboost creates several weak classifiers h_j, each of these evaluates a Haar characteristic j over an image x_j and through the comparison between the obtained value from the evaluation and a threshold θ, it decides if this characteristic represents effectively the interest object. A weak classifier is defined by Equation (1).

$$h_j(x) = \begin{cases} 1 & f_j(x) < \theta_j \\ 0 & f_j(x) \geq \theta_j \end{cases} \quad (1)$$

Adaboost will find the best threshold and the best classifier through linear searches and reweighting of the examples with highest classificacion error ε_j, thus maximizing the margin between a positive and negative set of examples (x_j, y_j), being y_j1 or 0 for positives and negatives examples respectively. The classification error is defined by Equation (2).

$$\varepsilon_j = \sum_i w_i |h_j(x_j) - y_i| \quad (2)$$

Then Adaboost will use the best classifier to create a combination that has better discrimination accuracy, this combination is called strong classifier h and is defined by Equation (3).

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha h_j(x) \geq \sum_{t=1}^{T} \alpha \\ 0 & \sum_{t=1}^{T} \alpha h_j(x) < \sum_{t=1}^{T} \alpha \end{cases} \quad (3)$$

Where $\alpha = -\log \beta_t$ & $\beta = \varepsilon_j / (1 - \varepsilon_j)$

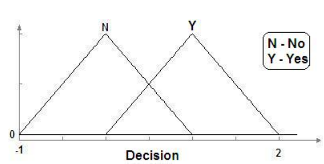


Figure 7. The Fuzzy Set of *Decision*

Table IV
PERFORMANCE COMPARISON BETWEEN DIFFERENT SIZES OF INPUT PATTERN.

Input pattern size	Hit rate (%)	False positives (%)	False negatives (%)
18x18	0.83447099	0.24420402	0.16552901
20x20	0.7440273	0.11919192	0.2559727
24x24	0.71331058	0.43206522	0.28668942
20x18	0.74573379	0.14145383	0.25426621

For the present work, the positive examples set (x_j, y_j) being $y = 1$ is extracted from traffic videos of several points of the city. From these videos 6364 images are obtained, for each one of these images true regions are annotated; 10050 true regions were found, thus obtaining the same number of positive examples. In order to obtain the negative example set, videos from daily scenes of parks and walkways are used, besides the image datasets from Google, CALTECH, CMU, TU Darmstadt, UIUC, VOC2005 y TU GRAZ are used too, from these, 8131 images are extracted in which do not exist a single car with frontal view.

The performance of the whole object detection system depends on several training parameters of the strong and weak classifiers, as well as the cascade itself, some examples of these parameters are: the size of the example sets, number of stages of the cascade, type of weak classifier... etc. In order to estimate the optimal values for these parameters, a series of experiments based on the work of [7] were conducted, but for the specific case of vehicles as interest objects.

In order to carry-out these experiments, a sub-set test was extracted from the positive example set, consisting of 152 images which contain 586 vehicles (likewise 586 true regions are annotated). These vehicles fulfills the criteria to be considered interest object, having a frontal or top frontal view and a maximum rotation from the frontal view of 30° just as Figure 8 shows.

Figure 8. Terms of the car position, to be considered an object of interest

Then, the cascade with the evaluated parameters is used to obtain new true regions, and these ones are compared to those previously annotated. The criteria for true positive and false positive are determined by two difference margins between new true regions and previous ones. One margin is for size, and has a maximum difference between each other of 50%. The other one is for location, with a maximum difference of 30%.

Table IVshows the influence of pattern training size in the performance of the cascade. The size patterns which obtains better performance (less false positives and a higher hit rate) are 18x18 and 20x20.

Table V shows the influence of the numbers of features used to train the weak classifier on the cascade performance. Three types of weak classifiers are used: one with one feature

Table V
PERFORMANCE COMPARISON BETWEEN THE NUMBER OF FEATURES PER WEAK CLASSIFIER

Weak classifier	Hit rate (%)	False positives (%)	False negatives (%)
Stump	0.83447099	0.24420402	0.16552901
CART2	0.73720137	0.13253012	0.26279863
CART4	0.69624573	0.12258065	0.30375427

(Stump), and two with 2 and 4 features respectively (CART). CART classifier with 2 features (hereinafter known as CART2) has shown the best overall performance, since it presents a reduction of 45% in false positives compared to Stump classifier, and only a 11% reduction in the hit rate.

The influence of Haar features set types, can be observed in Table VI. There are two types of Haar features sets, the basic one, proposed in [18] and the extended one, proposed in [7]. Extended set has shown a reduction on the presence of false positives up to 71% compared to the basic one, with a similar hit rate.

During the training phase, it is assumed that vehicles are symmetrical regard vertical axes, but what if this assumption is omitted? Probably, the detector could be more robust against rotations of the vehicle view. Table VII shows that this assumption is not valid, since the cascade without symmetry presents a drop in the detection rate, with little improvement on the insensitivity to false positives.

Based on the observations made in the previous experiments, the values for the final training parameters are shown in Table VIII.

Besides, based on the literature, other parameters were established e.g. the number of cascade stages must be between 15 and 23 stages, and the size of the training sets should be of about 5000 positives examples and 10000 negatives examples.

IV. EXPERIMENTAL FRAMEWORK

In order to verify system performance in a controlled, but projectable environment, it was necessary to implement a test scenario using artificial videos. For this, an algorithm was developed using MATLAB, which is capable of creating random videos that simulate traffic flow in a lane (see Figure 9a).

Table VI
PERFORMANCE COMPARISON BETWEEN THE TWO EXISTING SETS OF HAAR FEATURES

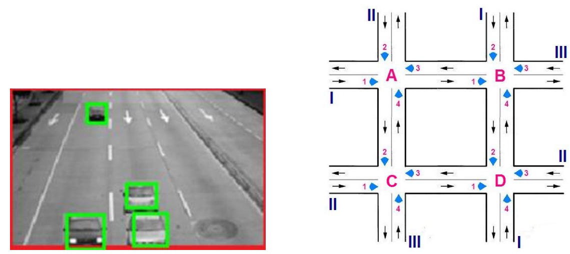
Feature set	Hit rate (%)	False positives (%)	False negatives (%)
Basic	0.84300341	0.35085414	0.15699659
Extended	0.83447099	0.24420402	0.16552901

Table VII
PERFORMANCE COMPARISON BETWEEN ASSUMPTION OF VERTICAL SYMMETRY

Vertical symmetry	Hit rate (%)	False positives (%)	False negatives (%)
With	0.77133106	0.22068966	0.22866894
Without	0.83447099	0.24420402	0.16552901

Table VIII
TRAINING PARAMETERS

Training parameter	Parameter value
Input pattern size	18x18 and 20x20
Weak classifier type	CART2
Features set	Extended
Vertical simmetry	Yes
Relationship between training sets	1:2 (Positive : Negative)



(a) Artificial created frame (b) Scenario diagram

Figure 9. Test scenario designed

As shown in Figure 9b, the designed scenario includes 4 simple two-way intersections, therefore a total of 16 videos were created representing each of the pathways of interest.

In order to compare the performance of the developed system over fixed-time controllers, both of them were tested under the same traffic conditions. Ten evaluation plans were designed and each of them varies the level of congestion on the tracks as shown in Table IX. This level of congestion depends on the type of the lane; Figure 9b shows that there are three types of lanes (I, II and III) and also shows the distribution of these types between the available lanes.

Table IX
GENERATION PLANS

	Type		
	I	II	III
P1	Very High	Low	Medium
P2	Medium	High	Low
P3	Low	Medium	High
P4	Very High	Very High	Low
P5	Very High	Medium	High
P6	Medium	Medium	Low
P7	Medium→High	High→Very High	Medium→Low
P8	Low→High	High→Very High	Medium→High
P9	Low→Medium	Medium→High	Medium→Low
P10	High→Low	Medium→High	High→Medium

V. RESULTS

For each controller (Fixed-time and Fuzzy), each of the plans was executed for 20 minutes. In order to compare the performance of each of them, two control variables were evaluated: the first one was the average delay time of each of the simulated vehicles, and the other was the number of cars that each controller was able to handle in the same period of time.

According to Table X, the results show that the developed system reduces the time delay caused by unnecessary stops in about 20%. It is also important to note that the system was able to adapt quickly and efficiently in those plans where there was a change in the level of congestion (7, 8, 9 and 10), outperforming the standard controller up to 26%.

On the other hand, Table XI shows that the controller developed was able to deal with a number of vehicles much larger than the standard controller, improving performance by up to 28.45%.

Finally, as explained in Section III-B - Table VIII, two identical classification cascades were created, the only difference between them was the size of the input pattern. In this way Cascade No. 1 has a size of 18x18 pixels while Cascade No. 2 has a size of 20x20 pixels. The results obtained with each cascade are shown in Table XII.

Besides both cascades presented similar performance in terms of processing speed, reaching a detection rate between 22 and 27 frames per second on images of 320x240 pixels.

Table X
AVERAGE DELAY TIME FOR EACH CONTROLLER

	Delay Time (%)		Imprv. (%)
	Fixed-Time	**Fuzzy**	
P1	58,94	49,46	16,08
P2	59,01	44,79	24,09
P3	56,22	42,82	23,83
P4	64,54	58,64	9,14
P5	64,81	56,01	13,57
P6	57,36	26,45	26,45
P7	62,92	50,98	18,98
P8	60,62	51,69	14,72
P9	58,68	43,35	26,13
P10	61,21	47,19	22,9

Table XI
NUMBER OF SIMULATED VEHICLES FOR EACH CONTROLLER

	Average number of simulated vehicles		Imprv. (%)
	Fixed-Time	**Fuzzy**	
P1	1251	1543,6	23,38
P2	1284,8	1483	15,42
P3	1231,8	1348,6	9,48
P4	1400,2	1779,6	27,09
P5	1379,2	1765,6	28,01
P6	1240,2	1342,6	8,25
P7	1345,6	1635,8	21,56
P8	1265,2	1625,2	28,45
P9	1270,6	1424,6	12,12
P10	1330	1541,4	15,89

Table XII
PERFORMANCE OF THE CLASIFFICATION CASCADES DEVELOPED

Cascade	Hit rate (%)	False positives (%)	False negatives (%)
No.1	0.8830	0.04381	0.117
No.2	0.8641	0.08143	0.1359

VI. CONCLUTIONS

The created vehicle detector is robust against several kinds of noise, just like moderate lighting variations, shadows, reflections and other types of phenomena caused by climatic condition. This advantage puts the chosen method above others, like background estimation and optic flow estimation.

Unlike vehicle detection methods based on optic flow calculation, the constructed detector is able to locate the vehicles even when these are stopped. In the same way, unlike methods based on *tripline* techniques, the constructed method does not show problems if vehicles change lanes intermittently or if these does not transit through certain predefined areas of the image.

On the other hand, the results show that the proposed controller's performance far exceeds that of fixed-time controllers, and also this can be optimally adapted to a large number of situations. Similarly, the project is applicable to any situation, regardless of the number of intersections or distribution.

Finally, it is observed that machine vision algorithm proposed for the detection of vehicles, presents a clear disadvantage, as is the lack of robustness to the presence of occlusions of the objects of interest, requiring that these occlusions are less than the 10% of the total area of the object. Therefore the location and height at which video sensors are installed should be considered, so that the level of occlusion between vehicles could be reduced.

REFERENCES

[1] J. R. Jorge Vicente. *Implementación de un sistema de simulación para el control del tráfico automotor*. Universidad el Bosque. Bogotá, Colombia. 2007.

[2] K. T. Kok, K. Marzuki, Y. Rubiyah. *Intelligent traffic lights control by Fuzzy Logic*. Artificial Intelligence Center. Universiti Teknologi Malaysia. *Malaysian Journal of Computer Science, Vol. 9 No. 2, Diciembre 1996, pp. 29-35.*

[3] L. J. LEE, K. *Distributed and Cooperative Fuzzy Controllers for Traffic Intersections Group*. IEEE Transactions on Systems, Man, and Cybernetics, Part: C. Vol. 29, No. 2. *1999.*

[4] ROBLES, Daniel, ÑAÑEZ, Pablo, QUIJANO, Nicanor. *Control y simulación de tráfico urbano en Colombia: Estado del arte*. Revista de ingeniería #29, Universidad de los Andes. Bogotá, Colombia. ISSN: 0121-4993. 2009.

[5] PAUL VIOLA, Michael Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*.

[6] RAINER, Lienhart, JOCHEN, Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection*.

[7] RAINER, L., KURANOV, A., VADIM, Pisarevsky. *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*.

[8] WIERING, Marco, VAN VEENEN, Jelle, VREEKEN, Jilles, KOOPMAN, Arne. *Intelligent Traffic Light Control*. Institute of Information and Computing Sciences. Utrecht University. 2004.

[9] SZELISKI, Richard. Computer Vision: Algorithms and Applications. September, 2010.

[10] CHOI, Jaesik. Realtime On-Road Vehicle Detection with Optical Flows and Haar-like feature detector.

[11] HAN, Sungji, HAN, Youngjoon, HAHN, Hernsoo. Vehicle Detection Method using Haar-like Feature on Real Time System.

[12] HIGASHIKUBO, Masakatsu, TOSHIO, Hinenoya, KOUHEI, Takeuchi. Traffic Queue Length Measurement Using an Image Processing Sensor.

[13] MICHALOPOULOS, Panos G. Vehicle Detection Video through Image Processing: The Autoscope System. 1991.

[14] FATHY, M., SIYAL, M. Real-time image processing approach to measure traffic queue parameters. 1995.

[15] WEBER, Ben. Generic Object Detection using AdaBoost.

[16] KLEIN, Lawrence A. Vehicle Detector Technologies for Traffic Management Applications.

[17] Y. Luz Elena, Y. Mimbela A. Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems.

[18] PAPAGEORGIOU, Oren. A general framework for object detection.