NIGHT VIDEO TRAFFIC DETECTION USING FREQUENCY FILTERS

Răzvan GHIȚĂ[1], Ana Maria Nicoleta MOCOFAN[2]

This article aims to present a study of the traffic video detection using different video processing methods capable of detecting the vehicles in poor illumination conditions, even at night. For achieving this target, a couple of spatial and frequency filters have been used. All the processes necessary to obtain the traffic flow detection have been concentrated and presented into a software diagram.

By combining this method with various other traffic detection methods we can compile a Lab View software application capable of being integrated with different intelligent transport management systems.

Keywords: video, spatial filter, frequency filter, peak, curve-fitting

1. Introduction

One of the non-intrusive technologies is represented by the video processing. It appeared in 1970's in USA. Between 1970 and 1980 several countries like U.K, Germany, France and Sweden tried to develop this technology [1].

Meanwhile the video traffic detectors have developed from the PC expansion boards to completely separated units dedicated for this process. The traffic video techniques have now a great advantage. They have the capacity to monitor a higher number of traffic lanes; they are flexible and feed a rich data stream into the traffic management system. Also, after years of research, the cost of this technology has been greatly reduced and the field experience greatly enhanced.

One of the challenges offered by the video detection consists of determining the traffic flow in poor illumination conditions during different periods of day, like night and dusk. The purpose of the papers is finding an algorithm suitable for the night traffic detection that will be included into an application specialized in traffic flow detection.

[1] PhD student, Depart. for Telematics and Electronics in Transports, Faculty of Transports, University POLITEHNICA of Bucharest, Romania. email:razvan_ghita_its@yahoo.com[2]

[2] PhD student, Depart. for Telematics and Electronics in Transports, Faculty of Transports, University POLITEHNICA of Bucharest, Romania.email: nicoleta_mocofan@yahoo.com

The paper presents a road traffic night detection algorithm developed by the authors using Lab View programming language. This algorithm will be used in combination with other motion detection algorithms to create a complete video detection application which will represent a cost effective solution for video based traffic management systems.

By means of this application, the images are captured and processed using the Lab View programming language. The vehicles entering the region of interest will be automatically counted. The traffic flow will be measured and this way it will be optimized with the aid of the traffic lights signal control. In the same time, a database containing the traffic flow data for each day and hour can be saved for future optimizing operations.

2. Using the region of interest for image extraction

In order to make possible the traffic flow detection during night, we need to enhance the usual traffic detection methods by using some pixel intensity manipulation techniques. We must take into consideration the fact that, during the night, the pixel intensities, which are building up the image histogram, are not keeping the same properties like in day time. Considering this, the value interval is limited, the hue is darker and the color of a vehicle cannot be distinguished from the color of the asphalt. In these conditions the only thing that can indicate the presence of a vehicle is represented by the headlights. The only drawback of this parameter is the lack of the vehicle classification ability.

In the following chapters a method of detecting the traffic flow during night will be presented. This method uses image processing techniques like frequency filtering and spatial filtering.

The frequency filters are used for altering the pixel values considering the periodicity and the spatial distribution of light intensity variation within an image [2]. By applying a high-pass filter we can isolate the templates that have a quicker variation and by applying a low-pass filter we can isolate the templates that have a gradual variation, like the objects from an image. To be able to apply a frequency filter to an image, the processed image must be transform into its frequency representation.

Before applying a filter, the image must be processed using the Fast Fourier Transform (FFT). The FFT can highlight information regarding the periodicity and the dispersion of templates found into the source image. The main principle of applying this transform is shown in fig.1 [2].

Fig. 1. Image processing using frequency filters [3].

The image processing techniques will be applied only to the desire portion of the image, in order to save processing power and hardware resources. In fig.2 we can see the block diagram section which is responsible for processing the extracted image portion and applying the necessary frequency filters for vehicle night detection.

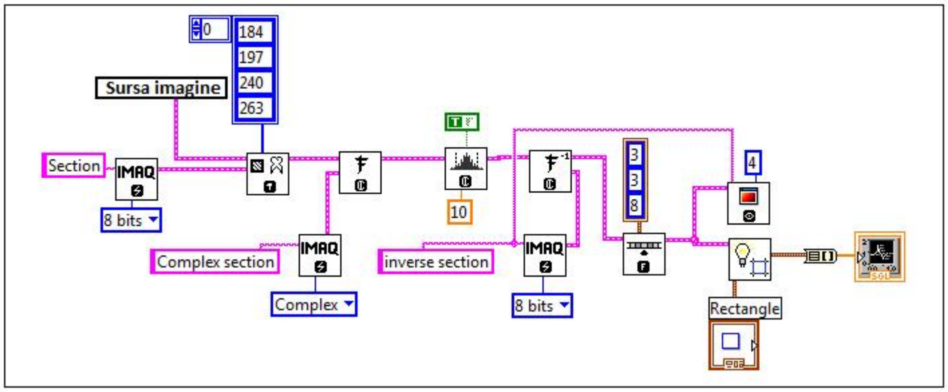

Fig. 2. The block diagram for applying the frequency filters [3].

The first step consists of extracting a portion of the image, from the original image, which is relevant for the traffic detection process. To accomplish this, the “IMAQ Extract” function will be used. This function will receive the parameters used for defining the region of interest. In this way the region of interest modification will determine an automatic modification of the extracted image section, upon which the night detection algorithm will be applied.

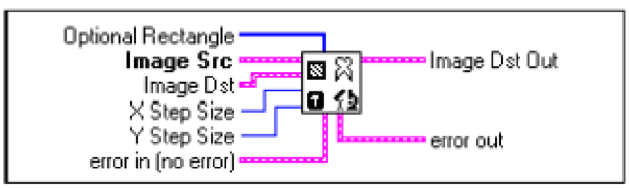

Fig. 3. The image extraction function [3].

The function presented in fig.3 is applied onto the original image which has already been the subject to an inversion function. The destination image will be defined by this function and it will be represented by an eight bit image. To define the limits of the extracted section the “Optional rectangle” field will be used. This field will receive the parameter of the region of interest previously defined by the user. When the user modifies the region of interest will also define automatically the section which will be processed. In fig.4 the transfer of the parameters from the region of interest to the extraction function it’s shown.

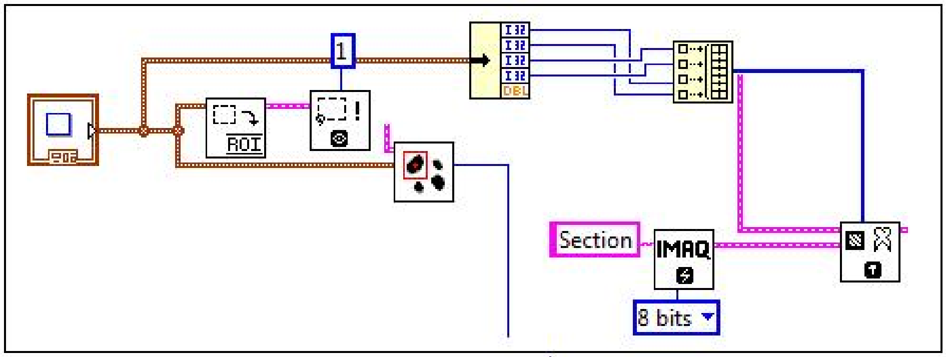

Fig. 4. The extraction function and the entry parameters [3].

As it can be seen, the parameters from the cluster that defines the region of interest are extracted according to their type (coordinates – integer 32, rotation - double). The parameters are introduces into an array that is used to feed these values into the extraction function. The result of this procedure is shown in Fig.5.

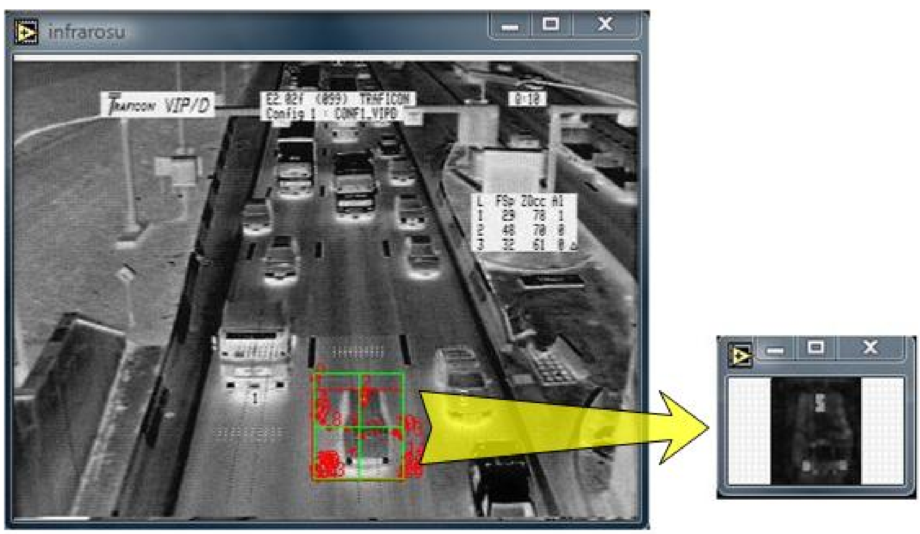

Fig. 5. The extraction function result.

Fig.5 represents a traffic video stream, from Traficon, which has been processed using the algorithm for night detection.

3. The standard representation of the Fast Fourier Transform

In this representation the high frequencies are grouped in the center of the image while the low frequencies are grouped on the edges (in the corners). The zero frequency is placed in the upper-left corner of the image. The frequency interval is presented below:

$$[0, N\Delta u] \times [0, M\Delta v]\ ; \qquad (1)$$

For a better perspective of the process the Fast Fourier Transform is applied here onto the entire image, not only for the region of interest. The FFT is applied only to the region of interest during the detection process, but here we'll use the entire image because the small size of the ROI is not too suggestive.

Fig.6 presents the image before and after it has been processed using the Fast Fourier Transform.

Fig. 6. The FFT standard representation (left) and the image before processing (right).

4. The optical representation of the Fast Fourier Transform

In the optical representation the low frequencies are grouped in the center of the image while the high frequencies are localized on the edges of the image. The zero frequency can be found in the center of the image. The frequency interval is presented below [4]:

$$\left[-\frac{N}{2}\Delta u, \quad \frac{N}{2}\Delta u\right] \times \left[-\frac{M}{2}\Delta v, \quad \frac{M}{2}\Delta v\right]; \qquad (2)$$

Fig. 7. The FFT optical representation (left) and the image before processing (right).

The optical representation is the one that will be used forward in the process and the one onto which the frequency filter will be applied. This representation is shown in fig.7.

5. Applying the frequency filter

After the image has been transformed into its complex representation, the next step is applying a high-pass frequency filter which has the role of truncating the low frequencies.

This filter eliminates a frequency f if it has values inside the truncation interval [f_0, f_c]. This is achieved by multiplying each frequency f with a C coefficient [3]. The frequencies outside the interval will keep the values from the original FFT complex image.

If $f < f_c$ then $C(f) = 0$, else $C(f) = 1;$ (3)

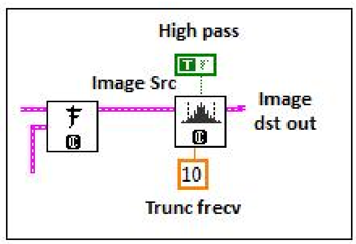

Fig. 8. The truncation function [3].

Fig.8 presents the frequency truncation filter. This has the role of isolating the high frequencies, allowing the vehicle detection at night by identifying their headlights. For the frequency truncation the following expression will be use.

$$f_c = f_0 + 10\%(f_{max} - f_0); \quad (4)$$

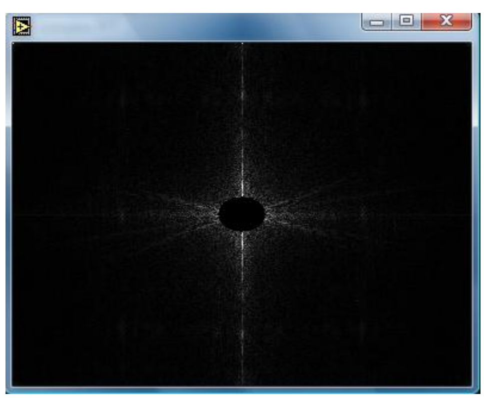

Fig. 9. The complex image after applying the high-pass filter.

After the frequency filter has been applied, an inverse Fast Fourier Transform is computed for the processed image (represented in fig.9). This has the role of converting the image back into its spatial representation to allow further processing by using spatial filtering [4].

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)e^{j2\pi(\frac{ux}{N}+\frac{vy}{M})}\,; \tag{5}$$

Fig.10 presents the result of processing a video frame by using a frequency high-pass truncation filter and applying the inverse Fast Fourier Transform for obtaining the spatial image. The truncation has the role of eliminating the reflection appeared on the camera lens or on the asphalt because of the light coming from the headlights of the vehicles passing on the road. These reflections can cause the appearance of some camera blinding effect by masking other vehicles from the near vicinity.

Fig. 10. The video stream before and after applying the truncation filter and FFT.

After the image has been processed using the Fast Fourier Transform, frequency filtering and brought back to its spatial representation, another filter, a spatial one, can be applied. Applying this filter represents the third stage of image processing for traffic flow detection during night.

6. Fine tuning the detection process by using spatial filtering

A spatial filter is used for altering a pixel value considering the variations of the light intensity in the pixel vicinity. This vicinity is given by an array which has as the center the same pixel. The purpose of this filter is to highlight the templates present in the given image, in this case the headlights of the vehicles and reducing the noise or other insignificant objects that are not important for the detection process.

If $K_{(i,j)}$ represents the light intensity of the K pixel which has the coordinates *(i,j),* the surroundings pixel intensities can be represented using the following array (in this particular case is a 3x3 array) [5]:

$$\begin{matrix} K_{(i-1,j-1)} & K_{(i,j-1)} & K_{(i+1,j-1)} \\ K_{(i-1,j)} & K_{(i,j)} & K_{(i+1,j)} \\ K_{(i-1,j+1)} & K_{(i,j+1)} & K_{(i+1,j+1)} \end{matrix} \quad (6)$$

For this application we'll use a linear filter called the "N'th order filter". This filter represents an extension of the median filter. It gives to each pixel the N'th value of its vicinity, when these values are arranged into an increasing order. "N" represents the order of the filter and can be used for granulating the filter effect on the image light intensity. If "N" is small, the image resulted after the image processing will be much darker.

Considering an image with the dimension 3x3, represented by the matrix (11) we can define the set M_{pixel} which contains all the elements of this matrix. The following expression will be valid:

If $M_{pixel}=\{K_{(i-1, j-1)}, K_{(i , j-1)}, K_{(i+1, j-1)}, K_{(i-1, j)}, K_{(i , j)}, K_{(i+1 , j)}, K_{(i-1 , j+1)}, K_{(i , j+1)}, K_{(i+1 , j+1)}\}$ and $N \in \{0,\ldots,8\}$, called "the order of the filter", then $\exists\ K_n$,called "the value of the new pixel", with the following values:

$$N = 8 \implies K_{n1} = \max[M_{pixel}] \quad (7)$$

$$N = 7 \implies K_{n2} = \max\left[M_{pixel} - \{K_{n1}\}\right] \quad (8)$$

$$N = 6 \implies K_{n3} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}\}\right] \quad (9)$$

$$N = 5 \implies K_{n4} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}, K_{n3}\}\right] \quad (10)$$

$$N = 4 \implies K_{n5} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}, K_{n3}, K_{n4}\}\right] \quad (11)$$

$$N = 3 \implies K_{n6} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}, K_{n3}, K_{n4}, K_{n5}\}\right] \quad (12)$$

$$N = 2 \implies K_{n7} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}, K_{n3}, K_{n4}, K_{n5}, K_{n6}\}\right] \quad (13)$$

$$N = 1 \implies K_{n8} = \max\left[M_{pixel} - \{K_{n1}, K_{n2}, K_{n3}, K_{n4}, K_{n5}, K_{n6}, K_{n7}\}\right] \quad (14)$$

$$N = 0 \implies K_{n9} = K_{(i,j)} \quad (15)$$

In this particular case a N=8 order filter has been used. Concerning this fact, the filter will have the tendency to dilate the bright particles and to sharpen the resulted image, making the vehicle detection process easier. In fig.11, the image before and after processing, can be seen.

Fig. 11. The image after applying the N’th order filter.

The image processed using the spatial filter has the headlights of the vehicles better delineated and dilated in order to deliver, at the end, a better defined set of values, which will help identifying more precise the passing of a vehicle through the region of interest. In fig.12 the set of the outputted values are represented.

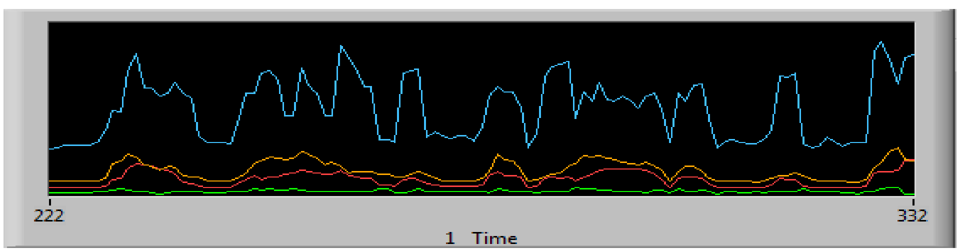

Fig. 12. The graphic of the pixel intensities.

The values from fig.12 have been obtained by applying the “light meter” virtual instrument for the region of interest defined earlier. The function processes the image after applying the FFT and N’th order filter on the video stream.

These values will make the object of the vehicle metering process. For achieving this goal a peak detector will be used, but after the signal has been mediated. The mediated signal is presented din fig.13.

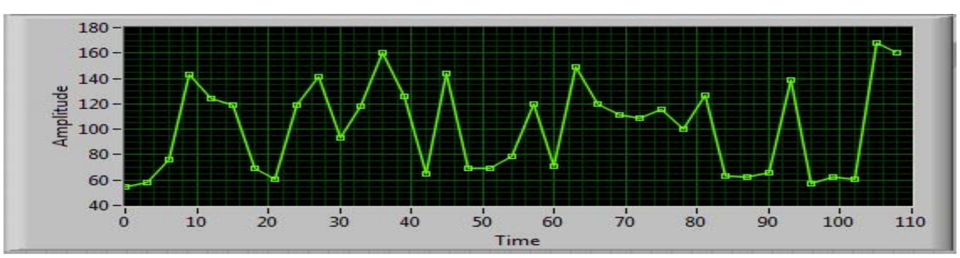

Fig. 13. The mediated values.

This mediation helps us to better identify the pattern of a vehicle passing the region of interest. Therefore, the peaks are signaling the presence of a vehicle.

After the signal has been mediated, a square curve fitting algorithm will be applied by the means of a peak detection function.

The peak detection function will give the final number of vehicles passing through the region of interest. Concerning the fact that the process will take place during the night, the classification of the vehicles won't be possible.

For emphasizing the quality of the values given by the image processing algorithms, below, the result of a 6 polynomial fitting, is presented in equation (16). The polynomial fitting uses the values presented in fig.13. The statistical factor R^2 will give the probability of detecting the peaks and also the probability of detecting the vehicles from the given set of values.

$$y = 1e^{-8}x^6 - 3e^{-6}x^5 + 0{,}000x^4 - 0{,}008x^3 - 0{,}027x^2 + 5{,}196x + 56{,}13. \quad (16)$$

$$R^2 = 0{,}295\,; \quad (17)$$

The statistical factor for the night detection method is ranked in second place when considering other detection methods like: dynamic method, template method or edge detection algorithm (all being vehicle detection algorithm used during day time). The best method is represented by the "dynamic method" which has a statistical factor R^2=0.437 (the ideal case being R^2=1), while the last is the edge detection with a factor R^2=0.233.

The processes, which are used in order to achieve the traffic flow detection during the night, are presented in fig.14 by using a software diagram. They consist of: video stream acquisition; extracting the region of interest section from the original image; Fast Fourier Transform computing; applying the frequency high pass filter; applying the N'th order filter and determining the traffic volume.

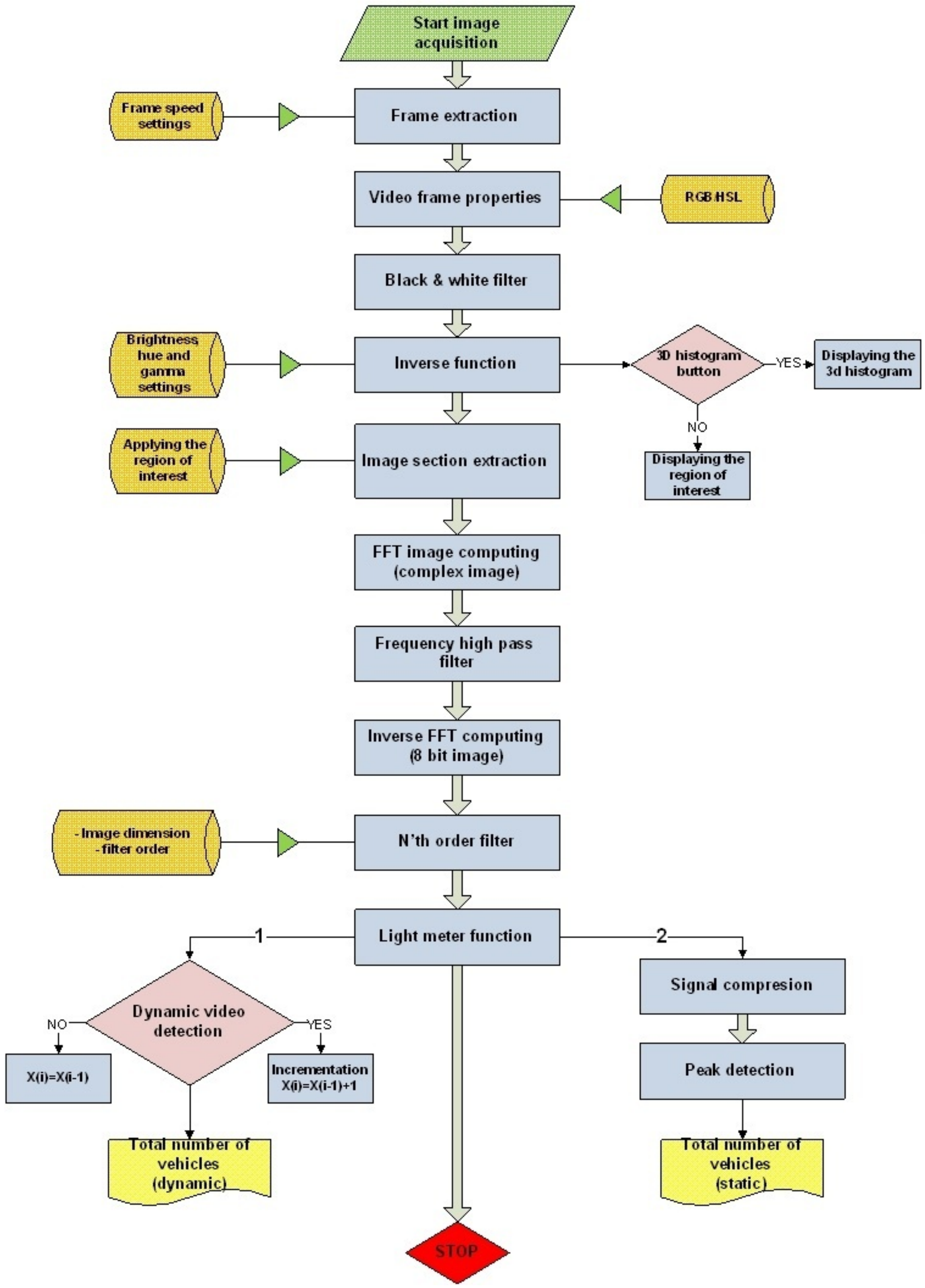

Fig. 14. The software diagram.

7. Conclusions

The main purpose of this paper is to present a new application, for the Lab View video processing capabilities, in traffic management systems. A main asset of a Lab View traffic video detection software is represented by the fact that this application eliminates the need of dedicated video processing hardware, like video boards or dedicated video cameras enforced by the producer in a package with the data acquisition boards.

To obtain a better traffic detection application, this method can be combined with different day detection algorithms. Also, combined with some network protocols fitted for video streaming, we can integrate this application with some traffic management or control system.

The approach proposed is not representing a genuine real-time detection method. It will accomplish the vehicle count after a pre determined time, which concurs with the traffic light cycle duration. Therefore, the number of vehicles will be provided after the cycle has ended.

A real-time vehicle flow determination is not necessary for this application because the traffic management plans will be built based on the data obtained from the previous cycles. Using this approach, the complexity of the software application is decreased and, in the same time, the cost of the final system is also diminished.

R E F E R E N C E S

[1] *Luz Elena Y. Mimbela and Lawrence A. Klein.* A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems, the Vehicle Detector Clearinghouse, New Mexico State University, Fall 2000

[2] *National Instruments,* Imaq Vision for G, Iunie 1997.

[3] NI Vision 2010 for LabVIEW Help, Edition Date: June 2010.

[4] *Thomas Klinger*, Image Processing with LabView, 11 Iunie 2003

[5] *Frank Y. Shih,* Image processing and pattern recognition, IEEE Press, 2010