



Răzvan Ghiță

Programming Methods for Road Traffic Control Video Applications

This paper aims to present an application capable of detecting the road traffic load in urban areas using an infrastructure of video cameras in conjunction with a well known programming language: Lab View. The presentation will highlight the role played by video detection technology, what advantages it offers and some examples of its successful use in the area of public transports. Developments in recent years clearly show that video systems certainly can, and will, establish themselves in the market. Also, an important aspect of the application consist of the advantage that it can determine the degree of occupation of a traffic link at certain hours, but it also can realize a traffic optimization by interconnecting this application with a traffic light adaptive, control system.

Keywords: road traffic load, video application, traffic optimization

1. Introduction

The system function will consist in road traffic monitoring with the help of a video camera infrastructure. The video cameras will be connected, using video acquisition boards, to the main traffic management servers, the information being transmitted via a fiber optic local area network. 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.

For a better traffic management it will be possible to create regions of interest which will generate an alarm for different incidents like accidents, illegal stops and parking, driving on dedicated bus or emergencies lanes, etc.

2. The front panel

The application consist of two parts, one is the user interface through which the operator can supervise the ongoing process and will be able to make minor modifications to the detection mode, and the block diagram which contains the source code and won't be available to the traffic operator. This block diagram will be available only for the programmer for improving the application, calibration (needed in uncommon areas which have particular properties), also for adding new controls and features that can be used by the traffic operator. In the following picture the user interface of the application is presented. Not all the features presented in this example are absolutely necessary for the operator to run his activity, these were added just for a better understanding of the process.

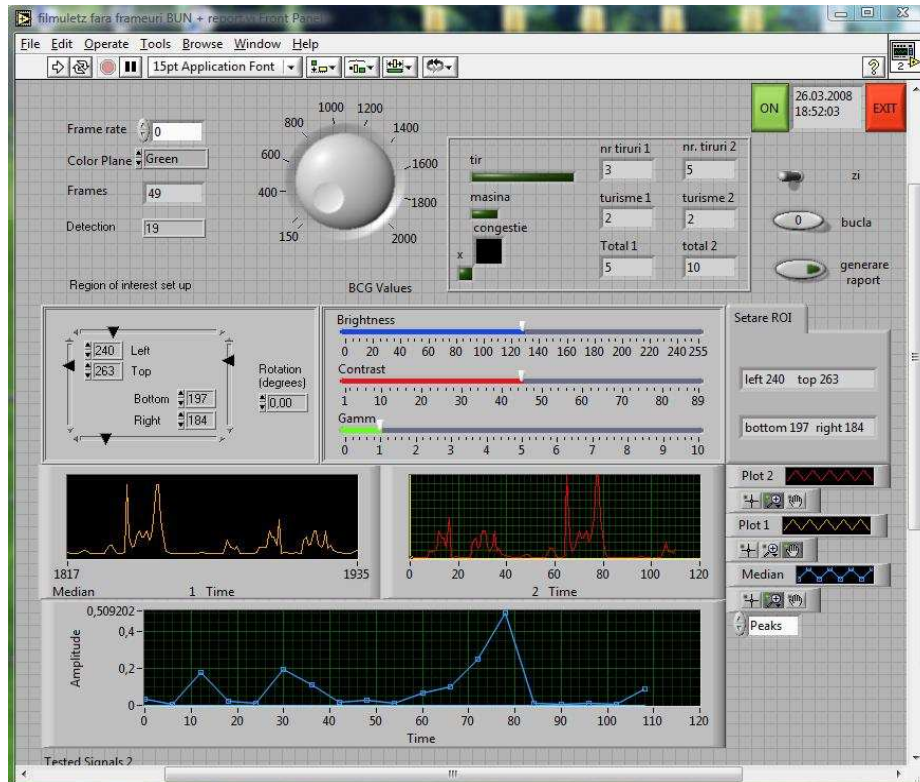


Figure 1. The user interface (Front panel).

The first group of controls/indicators that is shown in the front panel, is the group which allows the operator to view the number of the current frame (the one

displayed on the screen). We can find here the frame rate control button and also the indicator of the pixel value, detected in the region of interest. From here the image mode can be modified by selecting one from the following intensities options: green, red, blue or RGB. The button from the right side allows the modification of the frame time delay, resulting in this way a certain frame rate.

For optimizing the energy necessary for the video cameras to operate, and also for saving the bandwidth necessary for data transmission, an inductive loop will be simulated. This loop will detect the vehicle coming towards the intersection, and it will be active only during the night hours. When a vehicle travels on a link, it will be detected upstream by the inductive loop, the video camera will exit the stand-by mode and begins capturing and transmitting the images to the control center.

If the operator needs a better control for the detection process, the application can be provided with an optional set of controls that allows the setting of the region of interest (ROI) and also the modification of the luminosity and contrast. The region of interest can be placed in any desired area of an image obtaining, in this way, a better calibration. The ability to change the luminosity and the contrast values, allows us to obtain different values for the detected pixel. By doing this it will be possible to adjust the detection properties to match the requirements of a special area. In the image below we can see the group of controls in charge of this calibration.

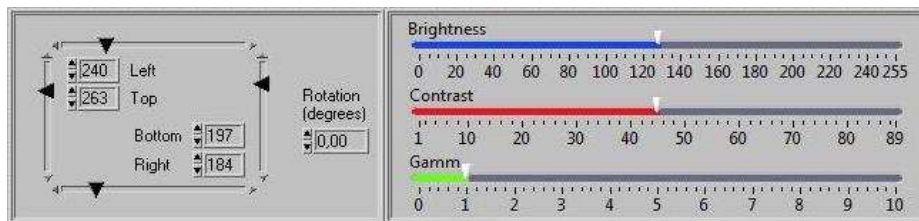


Figure 2. ROI and image setting controls.

After configuring the parameters the program is ready for the detection process. The number of the detected vehicles is displayed into a special group of indicators. This group is divided into two areas. The first area uses a dynamic method to count the vehicles, in this way the pixel values are monitored and a threshold is defined. The number of vehicles will be incremented when a pixel value that is in the range defined is followed by a value that is above the threshold defined. Using this method the vehicle number is incremented in real time while they are passing the detection area. The total number is determined and dynamically displayed. In the second area the vehicle detection method remains the same, but the detection of the heavy vehicles changes. They are determined after a certain number of vehicles have already passed, and displayed at certain time intervals, as we will be able to see in the following.

In addition to these controls the operator will be able to supervise the process with the aid of two video streams. One of them is used for traffic monitoring, and the other contains the region of interest for incident and vehicle detection.



Figure 3. The video stream (left) and the region of interest (right).

The region of interest will define the area that will supply the information to the detection blocks from the source code. These blocks will feedback the values of the pixel intensity to the front panel using charts and graphics like the ones in the pictures below.

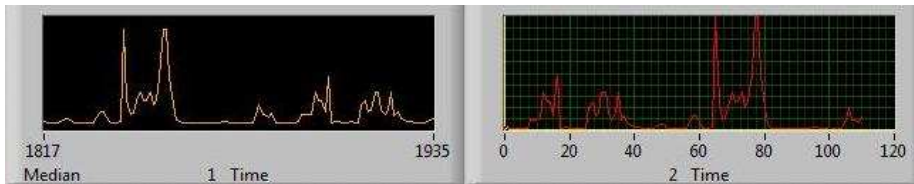


Figure 4. Chart showing the real time pixel values (left) and graphic (right).

The graphic shown above will be used in the heavy vehicle detection. The values from this graphic are tested for errors and all the values that reach a defined threshold (established by the programmer) are considered detection errors and eliminated. These values are not specific to any kind of vehicles and they appear completely random. The graphic that shows this value testing is shown below.

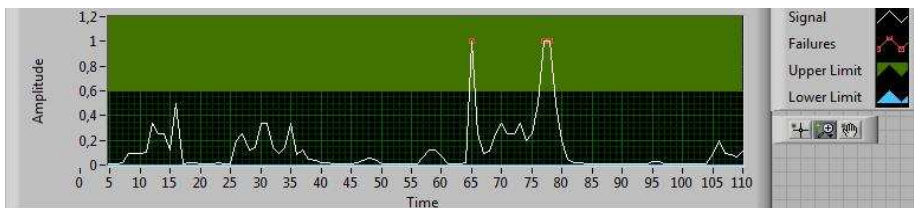


Figure 5. The value testing

Using this method the values over 0,6 are eliminated. The values that match the condition will be mediated and then used in heavy vehicle detection. Using these values another graphic will be drawn, and his peaks will determine the number of heavy vehicles that have been detected in the region of interest. In the following picture the graphic on which the peak detection has been applied is shown.



Figure 6. Peak detection graphic

After the configuration and acquisition of the detection data, the user can store all the data into a report that can be printed when needed. To print the report the path where this document will be saved must be entered into the field below.

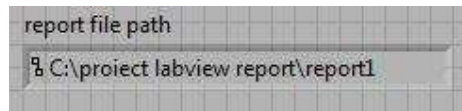


Figure 7. The report file path field

After launching the printing command a file that contains information about the vehicle number and classification will be stored on the hard drive. This file also contains the number of video frames that were transmitted, the pixel value and the region of interest settings.

3. The block diagram – image processing

The block diagram represents the source code that is hidden behind the front panel and manages the data stream that flows from the front panel to the block diagram and back. In this particular case the block diagram is divided into a couple of parts that perform different functions like image processing, vehicle detection, vehicle counting and classification and generation the report.

Bellow the first part of the block diagram is shown. This part is responsible for the image processing and detection.

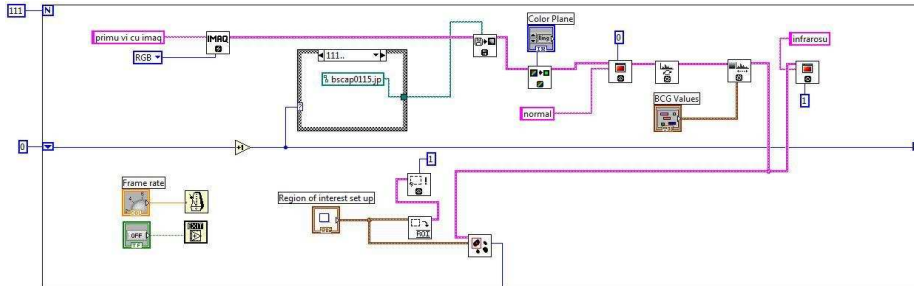


Figure 8.Image processing and detection

Lab View cannot handle large areas of memory, needed for image saving, that's why the image processing responsibility shifts to the IMAQ Vision module. A 8 bit image with a resolution of 512x512 fills 262,144 B or 265 KB of memory. The images are directly handled by the virtual instrument "IMAQ create". Each image must be associated with an unique name, which defines a generic structure representing all the aspects contained and associated with an image. An image structure may contain different information. This structure is interrelated with the image processing and the necessary functions.

This kind of image structure represents an entry for each virtual instrument (VI) and is also a specific data type which results directly from the execution of a "IMAQ create" VI. To be able to be executed, the virtual instrument must contain information about the image to be processed and the image that must receive the results.

To create an image the following procedure must be used.

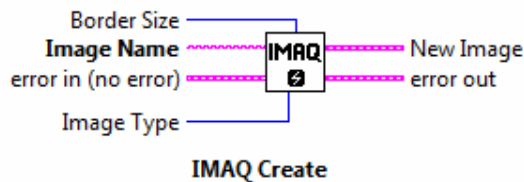


Figure 9.The Virtual instrument for image handling

Several images can be created by executing this VI a number of times equal with the number of images required. Each image created needs an unique number. The number of the required images can be determined from the application analysis. In this case the images are extracted one by one with the aid of a "FOR" instruction which has a number of iterations equal with the number of images that must be handled by the VI.

Another Vi is used to read the image: "IMAQ Read File". The image format can be a standard one (AIPD, TIF, BMP, JPEG) or it can be a non-standard format.

In each case the extracted pixels are automatically converted into the type of image that flows through the "image" entry filed.

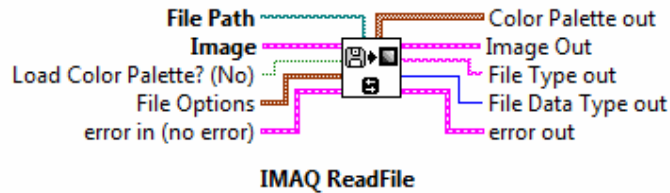


Figure 10.The Virtual instrument for image reading

For displaying the image the program uses a library which contains the virtual instruments capable of displaying the images in certain windows and also in charge of opening and closing this windows.

An example of such a virtual instrument is "IMAQ Wind Draw". This Vi displays the image in a window which automatically appears when the VI is executed. Usually the windows is not endowed with scroll bars, but these can be added using a different instrument.

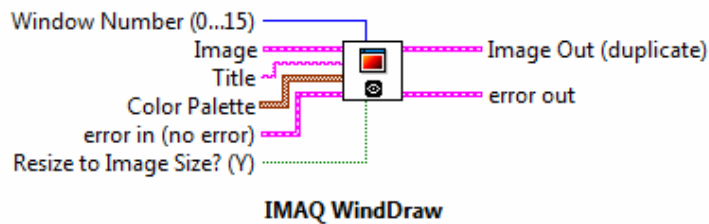


Figure 11.The Virtual instrument for image displaying

In what concerns the detection, it can be made by determining the intensity of the pixels which are located in a region of interest configured by the operator. This way the region of interest can be used to focus the image processing and analysis on a certain area of the image. The ROI can be configured using standard or free edges. The IMAQ Vision user has the following options:

- The association of a ROI with a window;
- The extraction of a ROI associated with a window;
- The deletion of an active ROI;
- The transformation of an image mask into a ROI.

An image mask which is converted into a region of interest, must be able to support an offset. The offset is used for placing the newly converted ROI into a space of another image.

The descriptor of a ROI (a cluster) contains the following elements:

- The margin rectangle;
- The list o regions, which contains a margin identifier (0 – for the exterior margin, 1 – for the interior margin);

- The type of the margin;
- The list of the points which describe the margins.

The virtual instrument used for associating a ROI with a window is shown in the picture below.

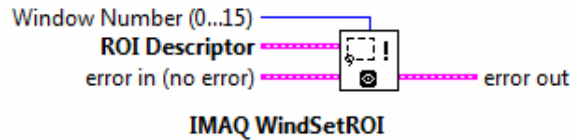


Figure 12.The Virtual instrument for ROI configuration

The window number (between 0 and 15) specifies the number of the window. The default value is 0.

For a better visualization of the pixel intensity evolution a geometric Vi called "IMAQ 3DView" must be used. This Vi can display an image using an anisometric view called a histogram. Each pixel is represented as a column in 3D view. The value of the pixel is correlated with the altitude of the column.

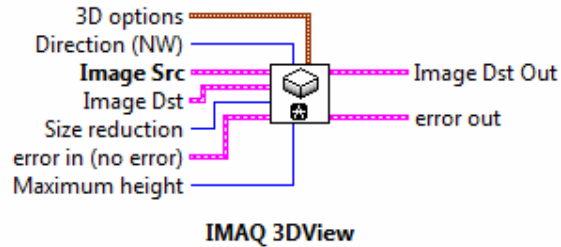


Figure 13.The Virtual instrument for Histogram

The entry cluster called "3D Options" has parameters like, "alpha" – which defines the angle between the base line and the horizontal line (0°- 45°), "beta" – which defines the angle between the horizontal and the second base line (0°- 45°), "border" – which defines the dimension of the frame in 3D view, and "background" – which defines the background color. An image 3D histogram is shown in the following picture.

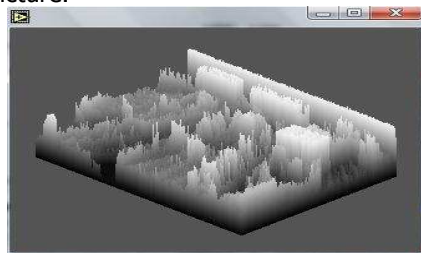


Figure 14.The 3D Histogram

An image histogram indicates the pixels quantitative distribution for each value of the grey level. It provides a general description of an image and helps to identify certain components like the background, the objects or the noise.

The histogram is a function H , defined in the grey level interval $[0, \dots, k, \dots, 255]$ for which the number of the pixels equal with the grey level value K is:

$$H(k) = n_k ; \text{ where:}$$

- K represents the grey level value;
- n_k the number of the pixels from an image that have a grey level value equal with k ;

The function of probability for the histogram is shown below:

$$P_{Linear}(k) = \frac{n_k}{n}$$

There are two types of histograms for an image: the linear histogram and the cumulative one. In both cases the horizontal axis represents the grey level interval. The vertical axis of the linear histogram indicates the number of pixels that have an intensity value of k , and the vertical axis of the cumulative histogram indicates the percentage of the pixels that have an intensity value less or greater than k .

The cumulative histogram has the following definition:

$$H_{cumul}(k) = \sum_0^k n_k ; \text{ where:}$$

- $H_{cumul}(k)$ indicates the number of the pixels that have an intensity value less or equal with k .

The function of probability for the cumulative histogram is :

$$P_{cumul}(k) = \sum_0^k \frac{n_k}{n} ; \text{ where:}$$

- $P_{cumul}(k)$ indicates the probability for the pixel's intensity to be less or equal than k .

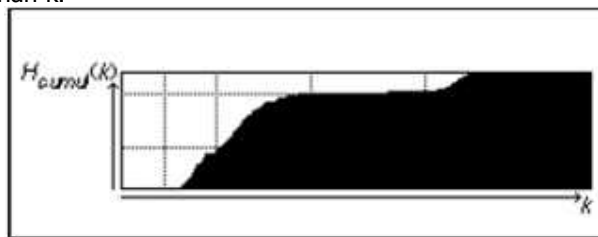


Figure 15.The cumulative Histogram

The grey level intervals which contains a concentrate set of pixels indicates the presence of a certain significant elements in the image but also the intensity

interval. The cumulative histogram shows that the image has some elements which represents 80% of it.

4. Case study – Bucharest traffic management system

This application can be used in traffic management systems that are not based entirely on video detection. A similar system is BTMS (Bucharest traffic management system) which uses an infrastructure of inductive loops for vehicle detection, but it already has a video camera infrastructure used solely for monitoring purposes. The problem of this systems consist of the drawbacks of the inductive loops solution, which includes the mounting and maintenance prices.

The images coming from the video cameras are supervised by the Bucharest police and allow the authorities to solve problems like traffic incidents or traffic rules disobeying by the traffic participants.

After a part of the system was installed, a month later:

- A person has been saved from his own car after he got sick. The incident was detected using the video camera infrastructure.
- Five minor accidents were discovered with the aid of video cameras, and the police took action in less than two minutes preventing the apparition ah other incidents or traffic jams.
- The travel time has been improved with 15% concerning that the traffic volume increased with 25% in the controlled areas (just 15 junctions were monitored when the statistic were made, today over 140 junctions are monitored).



Figure 16. The BTMS Control center

Some may think that video detection is not a mature technology in traffic control systems, but the fact that the video cameras proved to be a reliable solution, when they were tested in comparison with other technologies, makes them a real alternative to infrared or inductive loop sensors.

In BTMS systems were tested several solutions to find the one that fits perfectly with the city profile. The most efficient and precise solution proved to be the inductive loops, having the lowest error measuring rate from the entire set of detectors used. The next in line were the video cameras, placed far away from the stop line, the infrared ASIM detectors and in the last place was taken by the video cameras placed closer from the stop line.

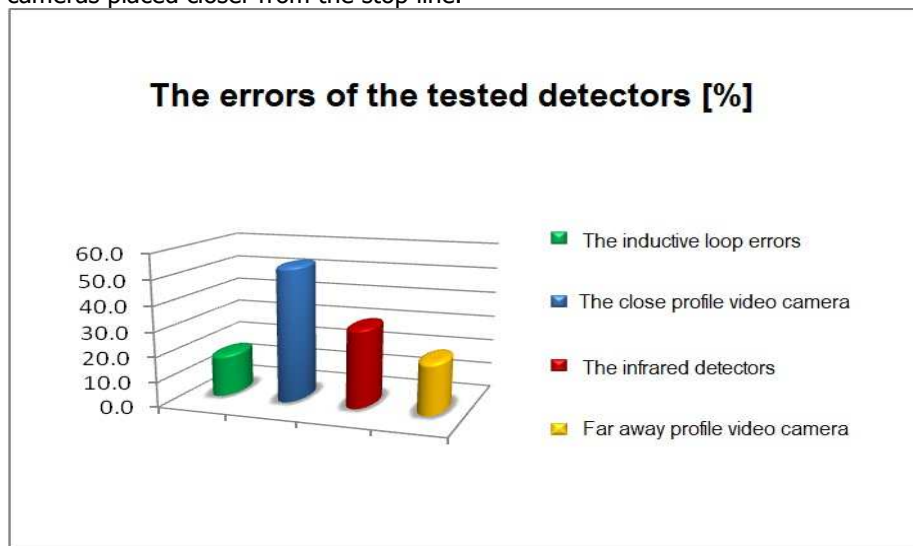


Figure 17. The errors of the tested detectors

5. Conclusion

The measurements result indicates the inductive loop as a proper method for vehicle detection. But we must consider the major drawbacks of this technology. One of them is the time lost with the traffic closing when it comes to mounting this type of detector in the traffic junctions. This requires a large amount of money and we also cannot neglect the traffic jams caused in other intersections by simply deviating the traffic on other links.

The low price of the equipment won't be felt because, in a city where the infrastructure is repaired at least once in three years, the installation and maintenance cost for the inductive loops will increase way over the price of a video camera infrastructure.

The video cameras have a higher equipment cost, but the maintenance cost is lower, the reliability is higher and the weather influence on the hardware equipment is way lower than in the case of the inductive loops.

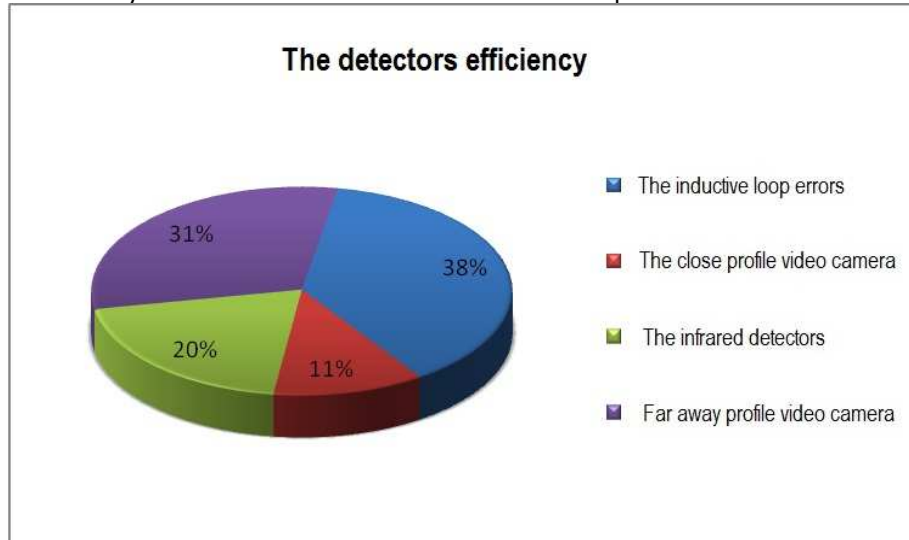


Figure 18. The detectors efficiency

Considering all these facts we can conclude that the video detection is the most reliable and precise vehicle detection method that can be used successfully in a road traffic management system.

References

- [1] National Instruments Corporation, *IMAQ Vision for G Reference Manual*. June 1997 Edition.
- [2] Russ J. C. *The Image Processing Handbook*, Third Edition. CRC, Springer, IEEE Press, 1999.
- [3] Drd. Eng. Razvan Ghiță, *Sensor network for road traffic management systems*, "Politehnica" University of Bucharest June 2008.

Address:

- Drd. Eng. Razvan Ghiță, "Politehnica" University of Bucharest, Splaiul Independentei nr.313, sector 6, Bucharest, razvan_ghita_its@yahoo.com