World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:10, 2016

Design and Implementation of a Counting and Differentiation System for Vehicles through Video Processing

Derlis Gregor, Kevin Cikel, Mario Arzamendia, Raúl Gregor

Abstract—This paper presents a self-sustaining mobile system for counting and classification of vehicles through processing video. It proposes a counting and classification algorithm divided in four steps that can be executed multiple times in parallel in a SBC (Single Board Computer), like the Raspberry Pi 2, in such a way that it can be implemented in real time. The first step of the proposed algorithm limits the zone of the image that it will be processed. The second step performs the detection of the mobile objects using a BGS (Background Subtraction) algorithm based on the GMM (Gaussian Mixture Model), as well as a shadow removal algorithm using physical-based features, followed by morphological operations. In the first step the vehicle detection will be performed by using edge detection algorithms and the vehicle following through Kalman filters. The last step of the proposed algorithm registers the vehicle passing and performs their classification according to their areas. An auto-sustainable system is proposed, powered by batteries and photovoltaic solar panels, and the data transmission is done through GPRS (General Packet Radio Service)eliminating the need of using external cable, which will facilitate it deployment and translation to any location where it could operate. The self-sustaining trailer will allow the counting and classification of vehicles in specific zones with difficult access.

Keywords—Intelligent transportation systems, object detection, video processing, road traffic, vehicle counting, vehicle classification.

I. Introduction

THE vehicle congestion is a common problem in any city that experiments a large growing in the automobile fleet without being able to go along with this growing by the road infrastructure. Due to the population increase and the number of vehicles in large cities, the traffic in the busiest streets and access roads tends to collapse, specially at the time with more circulation. The harmful effects of traffic jams are becoming more accentuated as the vehicular fleet grows, affecting the city economy and deteriorating the quality of life of their inhabitants. These effects can be a greater probability of accidents, the passage blocking of emergency vehicles and above all the increase in the time needed for arriving to destination. The last one causes other problems like an increase in fuel and vehicle maintenance expenses, air pollution, delay, road rage, less comfort during the trip and health problems. In order to face these problem it is necessary to take actions to reduce the vehicular congestion. This can be achieved by enlarging the road infrastructure, building new access and broaden the existing ones. However this leads to extremely high costs, so it is necessary to have an efficient vehicle traffic management that allows the optimization in the road usage. This can be done by the implementation of ITS (Intelligent Transportation Systems). Another reason of traffic congestion is the lack of information about the traffic conditions. The ITS offer tools that provides this type of information. One of the tools is the automatic vehicle counting, that obtains the number of vehicles going through a road in real time, and different methods have been used in order to achieved this, for example systems that use technology like inductive loop detectors, ultrasonic sensors, infrared, acoustic, radar detectors and video processing.

The systems based on video processing have many advantages in front of the other mentioned methods because they are non-intrusive, easy to install and maintained. However these systems tend to be fixed and operate only at the road crossing where they have been installed. The majority are powered through with main electrical power, which implies the use of external cabling and vulnerable to power outage, preventing their usage where this outages occur frequently.

This paper presents a system for counting and classifying the vehicles, capable of differentiating between automobiles, motorcycles or large vehicles like trucks or buses. The whole system will be implemented in a trailer, designed to facilitate their installation, maintenance and moving, coupling with some mean of transportation. The system will be self-sustaining through the use of three solar panels mounted over the trailer, which one of 120 W giving a total of 360 W, a gel battery of 140 Ah and a solar regulator of 30 A. As the system is not fixed, the trailer can be used to perform temporary measurements in different roads. The system will capture the traffic images with the help of IP cameras installed in a telescopic structure, which facilitates the installation and the cleaning of the lenses. The obtained images for each camera will be processed by a SBC Raspberry Pi 2, which has a 900 MHz quad-core ARM Cortex A7 CPU and 1 Gb of RAM memory. Also it will include a touch screen to perform the system control and to introduce manually the counting regions. The processed data will be stored locally in a database (MySQL 5.5.21) and they will be transmitted through GPRS packets for their storage in a remote Data Center.

Derlis Gregor and Kevin Cikel are with the Laboratory of Distributed Systems, Faculty of Engineering, National University of Asuncion, 2060 Isla Bogado, Luque, Paraguay.

Mario Arzamendia is with the Laboratory of Distributed Systems, Faculty of Engineering, National University of Asuncion, 2060 Isla Bogado, Luque, Paraguay (e-mail: marzamendia@ing.una.py).

Raúl Gregor is with the Laboratory of Power and Control Systems, Faculty of Engineering, National University of Asuncion, 2060 Isla Bogado, Luque, Paraguay.

II. Related Work

In order to detect objects in movement en captured videos from static cameras, one of the most employed techniques is the background subtraction. The principle of operation is based on the obtaining of the foreground objects through the removal of the background image. However the outdoor images present dynamic backgrounds, subject to variations due to changes in the illumination and meteorological phenomena, so it will be need to use a BGS algorithm that can adapt to these changes.

Benezeth et al. [1] compared the precision and recall of simple and complex BGS algorithms, in addition to obtaining their computational costs. The tests were performed in videos with multimodal backgrounds, i.e, that contain mobile parts, statics and with noise. Using as a reference a basic algorithm, which consist of simply using a static image as a background model and adapt it to compensate the illuminating changes, in order to calculate the average relative processing time for each algorithm, dividing the computational time of each one by the basic reference algorithm. The memory required was obtained calculating the minimum amount of floating point numbers required for pixel in each algorithm. It was observed that simpler algorithms like the basic algorithm and the Gaussian showed the less processing time and required memory, the GMM based algorithm showed the larger requirements, while more complex methods like the KDE (Kernel Density Estimation) showed very high computational costs, so it is not recommended in embedded systems application with limited resources and real time. In the videos with static background and without noise it was observed that all the algorithms present the same results. In videos with multi-modal background it is observed that the more basic algorithms are not capable of adapting to the background movements, while the videos with noise added to background the statistical methods, like the GMM, present the best results. Sobral et al. [2] developed the library called BGSLibrary based on OpenCV. This library provides functions in C++ for the implementation of 37 BGS algorithms.

As the vehicle shadows are also in movement, they are detected by the BGS algorithm, so it is necessary to employ shadow removal techniques to avoid identifying them as part of the same vehicle or as a new one. Sanin et al. [3] evaluate the detection methods of shadow based on physical, chromatic, geometric and texture features. They concluded that the methods based on geometric features are not adequate for different environments, but they are adequate when the objects are easy to model. The one based on chromaticism are the fastest to implement and execute, but are too sensitive to noise. The ones based on physical features improve the accuracy but fail if the spectral properties of the objects are similar to the background. The methods based on texture are the ones that present the more accurate results, but the computational costs are high.

Among the systems that perform the vehicle classification Chen et al. [4] developed a system for vehicle detection, tracking and classification, that uses a BGS algorithm based on GMM and SVM (Support Vectors Machine) for the vehicle classification, obtaining very good results.

Mithun et al. [5] developed a detection and classification system based on multiple time-spatial images, performing a vehicle classification in general categories through the analysis of shaped-based features, and then perform a more specific classification of the vehicle type by analyzing shape-invariant and texture-based features, obtaining very good results in the evaluations performed in different environments. In this case it is necessary to count on a set of training images for extracting the characteristics of different types of vehicles that will be used by the classifier. For a mobile counting and classification system for vehicles the distance, the location angle, and the circulation direction might change at each new location, so in order to use this classifier it will be need to have a set of training images for each new location.

Unzueta et al. [6] performed the vehicle classification using volumes in 3D that were estimated at the vehicle detection step, comparing the geometrical characteristics to perform the classification in two wheel vehicles, light vehicles and heavy vehicles. Lai et al. [7] classified the vehicles between trucks, buses and automobiles establishing threshold values of the relation of aspect and compactness. Sanchez et al. [8] classified the vehicles in the categories of large and small vehicles using the second entrance region separated from the first one by a distance dependent on the camera distance. If a vehicle passes through the first region is labeled as a small vehicle, and if passes through the second region before passing completely the first region, it will be labeled as a large vehicle.

III. Software Description

The software for counting and classification was developed using the computer vision tools from OpenCV 2.4.9 library and consist of four steps: image capturing, background subraction, detection and tracking, counting and classification. In 1, it is seen these step in a block diagram In 2 it is seen the sequence diagram from the main process, which includes all the software steps.

A. Image Capturing

At this step the image taken by the camera is captured and also the points that determine the counting zone. These points are aggregated manually for each new counting location, being stored in files so they remain available for the following software executions. The the ROI (Region of Interest) calculation is performed, where the counting will done from the regions defined. Finally the image is divided according to the number of lanes and all the zones outside the ROI are discarded. In this way the processing in the following steps will be performed independently for each lane, and it will prevent the waste of processing time from the zones outside the counting area.

B. Background Subtraction

This is the step that requires the most computational costs, so it is necessary to use an algorithm that it is not very complex to perform video processing in embedded systems with limited resources. The BGSLibrary 1.9.2 was used for

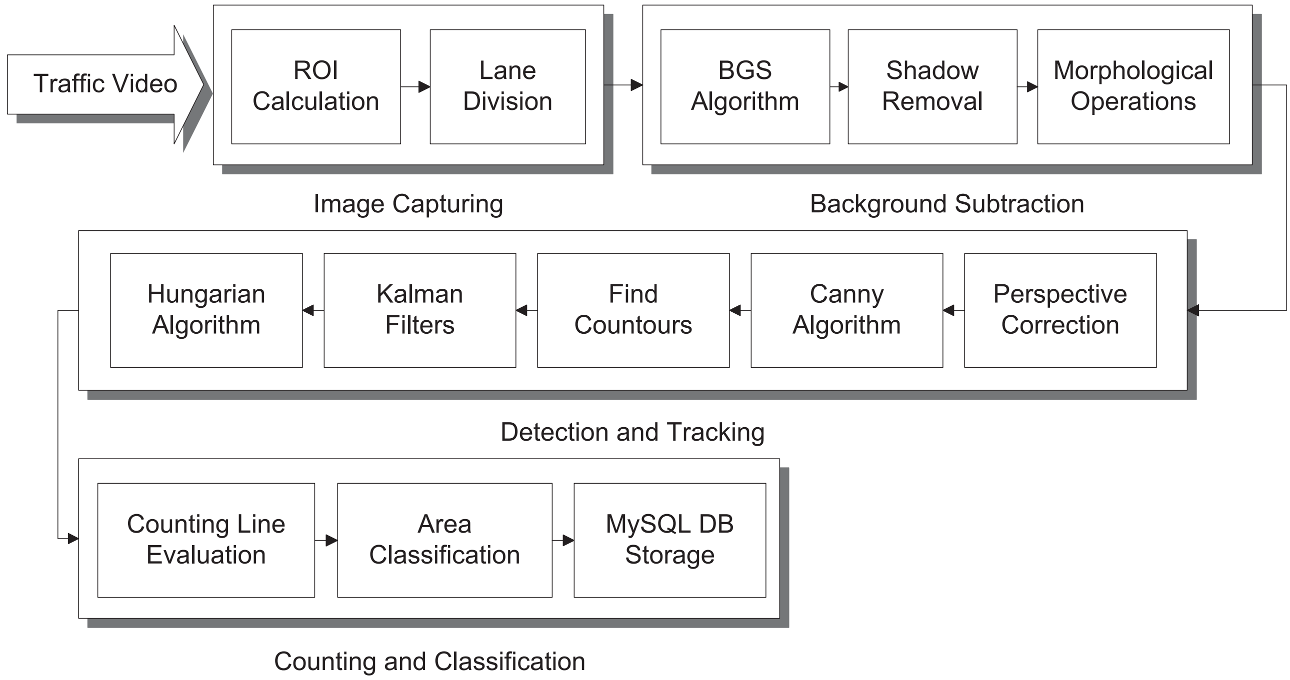


Fig. 1 Counting and classification software block diagram

implementing the Zivkovic GMM algorithm [9] based on the method developed by Stauffer and Grimson [10]. In this method, the most recent history of each pixel is modeled as a mix of K gaussians distributions, and the probability of observing the value of pixel X_t is described in:

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

where $\eta \left(X_t \mid \mu_{i,t}, \Sigma_{i,t}\right)$ is the Gaussian model number i and $\omega_{i,t}$ is the weight. $\Sigma_{i,t}$ is the covariance matrix that, for computational purposes, it can be assumed as a diagonal matrix in the form:

$$\Sigma_{i,t} = \sigma_{i,t}^2 I \tag{2}$$

Each new pixel value Xt is compared then with each one of the K existing Gaussian distributions, until finding one that matches. If none matches, the least probable distribution is replaced by a distribution with the average equal to the value of that pixel, with high initial variance and a low weight value. The average and the variance of the distribution that matches with the value of the pixel are adjusted in the following form:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{3}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho (X_t - \mu_t)^T (X_t - \mu_t) \tag{4}$$

where ρ is a second learning rate defined by the following form:

$$\rho = \alpha \eta (X_t | \mu_k, \sigma_k) \tag{5}$$

For the distributions that do not match, the variance and average values do not change. The distributions are ordered in a list based on the value ω/σ in a way that the most probable background distributions stay at the upper part of the list. In this ways the first list distributions are chose as part of the background model.

After applying the BGS algorithm, the shadows of the objects are removed by the shadow removal algorithm tha uses physics-based features developed by Huang and others [11]. In order to detect the shadows this algorithm uses an GMM to learn the color features of the pixels in movement. Then the gradient intensity distortion of each pixel is obtained, so the shadow differentiation with colors similar to the background is improved, employing for this a GMM for each pixel, which are updated through confidence-rated learning. Finally the posterior probabilities from the background, the shadows, and the foreground are obtained, and a pixel is assigned as part of the foreground if its posterior probability of the foreground is higher than the others.

Finally morphological operations of opening and closing are performed to remove small zones not corresponding to vehicles but detected as part of the foreground, and to fill holes that might appear in the vehicles.

Fig. 3 shows the sequence diagram of the function *subract()*, that relates to the BGS step, while Fig. 4, it can be observed the process performed in this step.

C. Detection and Tracking

In this step the foreground images obtained in the previous step are analyzed in order to detect and identify each one of the vehicles. At first the effects caused by the perspective are corrected, so the vehicles can keep a uniform size during the whole lane trajectory. In Fig. 5, the corrected perspective applied to both lanes in the original image is observed.

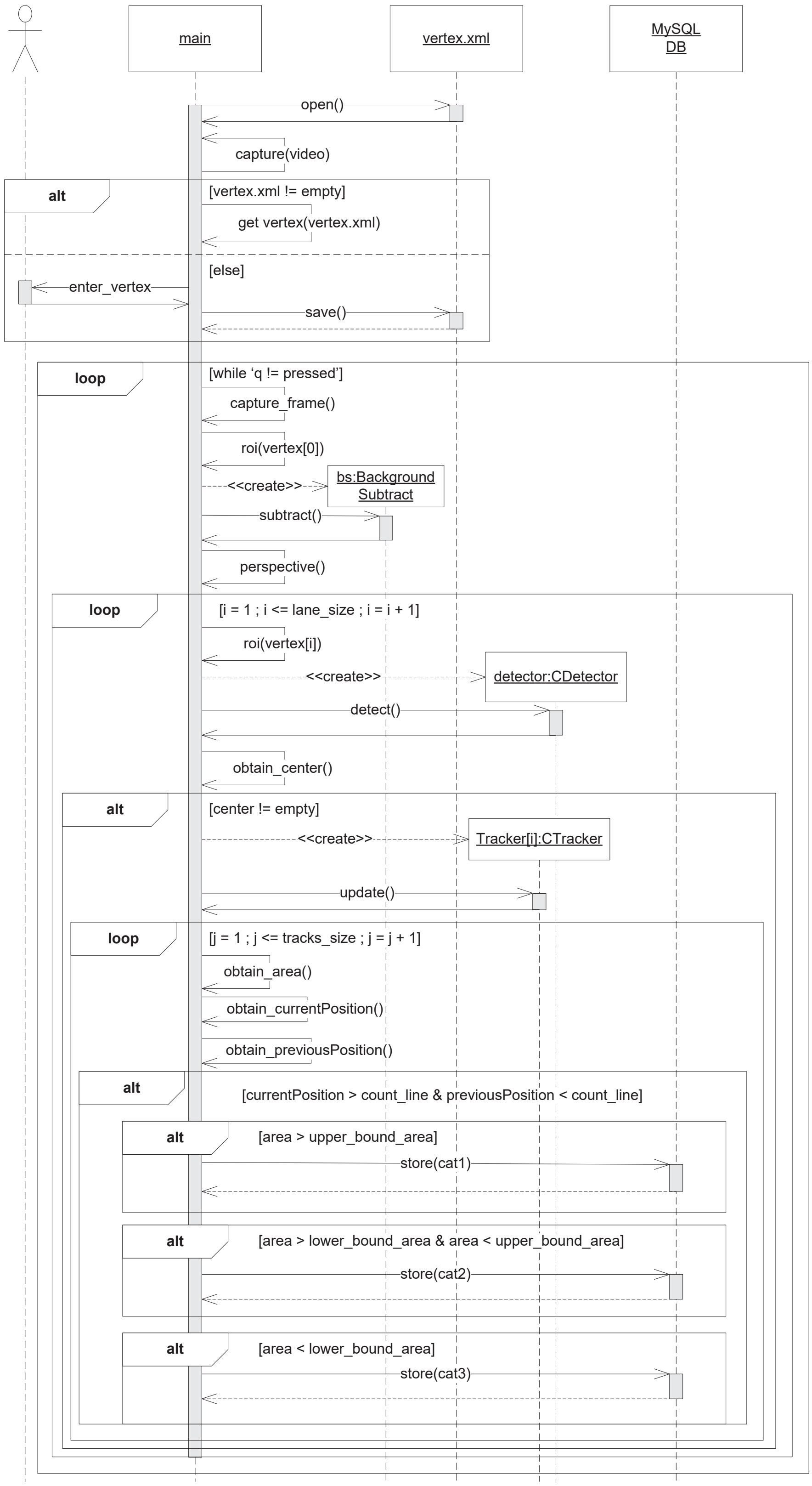


Fig. 2 Main process sequence diagram

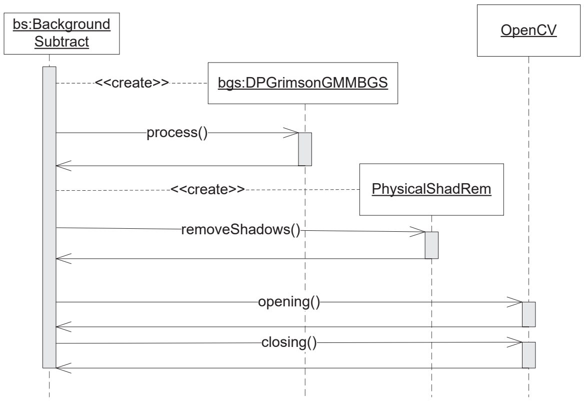


Fig. 3 Sequence diagram of the function *subtract()*

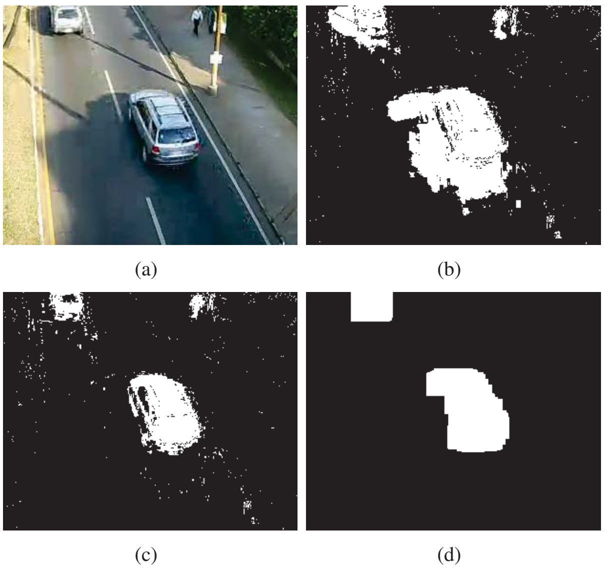


Fig. 4 BGS step. In (a) the original image is observed, in (b) the foreground image obtained through the BGS algorithm, in (c) the result after applying the shadow removal algorithm and in (d) the final result after morphological of opening and closing operations are performed

Fig. 5 Results obtained after applying the perspective correction. In (a) the counting regions defined in the step of image acquisition are observed. The perspective correction from both lanes are observed in (b) and (c)

After obtaining the location of each vehicle the edges for each one must be detected. The edge detection can reduce the quantity of information presented in the image, while the structural properties are preserved for later processing. For this the Canny edge detector is used [12], which consists of a multiple steps algorithm, and aims to meet three main criteria: low error rate, good localization and minimal response. In Fig. 6, the obtained edges by the detector are observed.

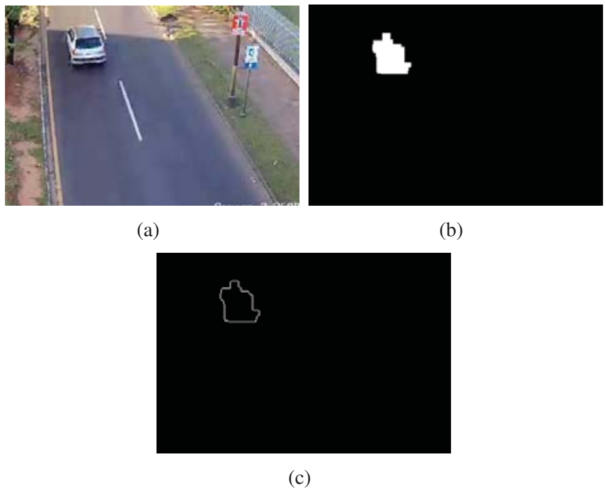


Fig. 6 Edges obtained by the Canny detector. In (a) the original image is observed, in (b) the output of the BGS step and in (c) the edges of the vehicle detected by the Canny algorithm

The find contours function of OpenCV based on the algorithm developed by Suzuki et al. [13] is used, that stores each edge found in the image as a vector of the points that compose it. These steps are performed through the function *detect()*, which sequence diagram can be seen in Fig. 7.

In order to perform the tracking of the detected vehicles the Kalman filters are used [14], that can estimate the future positions of the vehicles, according to previous positions of them, which allows keeping the tracking of a vehicle in case that the same stops being detected for a certain amount of time due to a vision obstruction caused by a larger vehicle. The Kalman filter uses an recursive algorithm that works in two steps: prediction and correction. In the prediction the future state estimated is generated, while in the correction

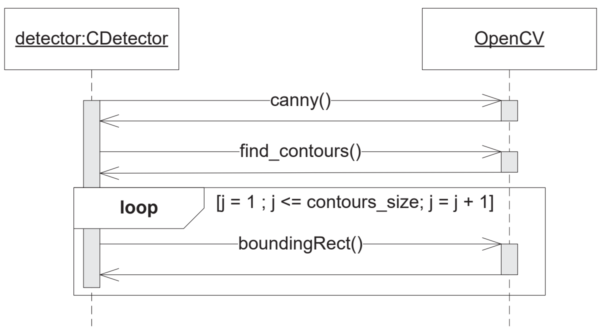


Fig. 7 Sequence diagram of fuction *detect()*

TABLE I
Testing the Counting Algorithm at Different CPU Frequencies in a Raspberry Pi 2

Frequency (MHz)	Video Time (s)	Execution Time (s)
900	180	1079
950	180	1029
1000	180	986
1050	180	946

TABLE II
Counting Test on a Video with Different Frames per Second Values

FPS	Video Time (s)	Execution Time (s)	Total Vehicles	Detected Vehicles	Counting Error (%)
30	330	1792	112	111	0.89
20	330	1449	112	110	1.78
15	330	1218	112	110	1.78
10	330	841	112	95	15.18
5	330	541	112	58	48.21

the error is statistically minimized , generating an improved estimation. Finally, the Hungarian algorithm [15] is used to do the assigment.

In Fig. 8, the sequence diagram of function is observed *update()*, which is in charge of performing the tracking and assignment.

D. Counting and Classification

After the tracking is performed, the current and previous location of each vehicle can be obtained, which allows the evaluation if the vehicle has crossed the counting line. In order to perform the classification, the minimum rectangle that contains the image of each detected vehicle is calculated, being able to determine the category (automobiles, motorcycles, truck or buses) that the vehicle belongs to according to the rectangle area. After detecting the crossing the counting is performed, registering at the MySQL database the time and date, the category in which the vehicle has been classified and the lane in which the registration occurred.

IV. Implementation

A. System Power

The system is powered by a 12V/140Ah battery. The calculated consumption of the entire system with two cameras is 5 Ah during daytime and 5.4 Ah during nighttime, the difference being due to the night vision infrared LEDs of the cameras. The cameras are PoE (Power over Ethernet) powered by networking cables. The Raspberry Pi 2 SBCs require 5V for power, so a powered USB hub capable of providing up to 2.4 A to each SBC is used. The power will be self-sustaining, using photovoltaic solar panels, which will be responsible for recharging the batteries. For controlling the level of battery, a voltage sensor module, connected to the Raspberry Pi 2 is used.

B. Video Capture and Processing

IP cameras were used to capture the traffic videos at a 360P (640x360) resolution, because at lower resolutions the differences in the areas of the vehicles types are reduced, resulting in high classification errors, and higher resolutions increase the computational costs. The algorithm is executed in Raspberry Pi 2 SBCs, using one for each camera. Tests have been made by running the counting algorithm at different CPU frequencies. The results are shown in Table I.

The lowest execution time was observed while running the algorithm at 1050 MHz, however at that frequency the Raspberry Pi 2 became unstable, because of that the CPU frequency was set to 1000 MHz.

The system must work in real time, so it is necessary to run the algorithm in parallel multiple times. The Raspberry Pi 2 has a quad-core CPU, that made possible to run simultaneously four process of the counting algorithm without increasing the execution time by running each process in a different core.

Table II shows the results of running the counting algorithm on a Raspberry Pi 2 in a traffic video with different frames per second (fps) values. The execution time and the counting error were tested. The lowest counting error was observed at 30 fps, but with a very high execution time, that make it unable to implement in a Raspberry Pi 2 for a real time execution. For that reason it has been decided to capture the images at 15 fps, that presented a similar counting error comparing to the test at 20 fps, but with a lower execution time that allows the implementation in a single Raspberry Pi 2 SBC.

C. Communication and Data Storage

It is used an 8-port NetGear GS108 Ethernet switch for the internal communication between the cameras and the Raspberry Pi 2 SBCs. The processed traffic data is stored locally in a MySQL database. For external communication a SIM800L GPRS module is used, which allows synchronization with a remote database, and also enables monitoring of battery level and system status.

D. System Control and Fault Recovery

The system has an additional Raspberry Pi 2 SBC that is connected with the modules and with a touchscreen display, whereby the counting regions are defined, and allows monitoring of the system. The system also has USB and network connectors, which will enable access to the internal network for control tasks, accessing via networking cables or using an USB wireless module. As an emergency system in case of system failure, there is a watchdog timer connected to a Relay, which will perform a reset automatically in case the system does not respond.

V. Conclusion

In this paper is proposed a trailer type system for counting and classification of vehicles through video processing. A four-step counting and classification algorithm was developed. At the first step the areas of the video to be processed are restricted. A GMM based BGS algorithm is used at the second step, because it is able to adapt to changes in the background

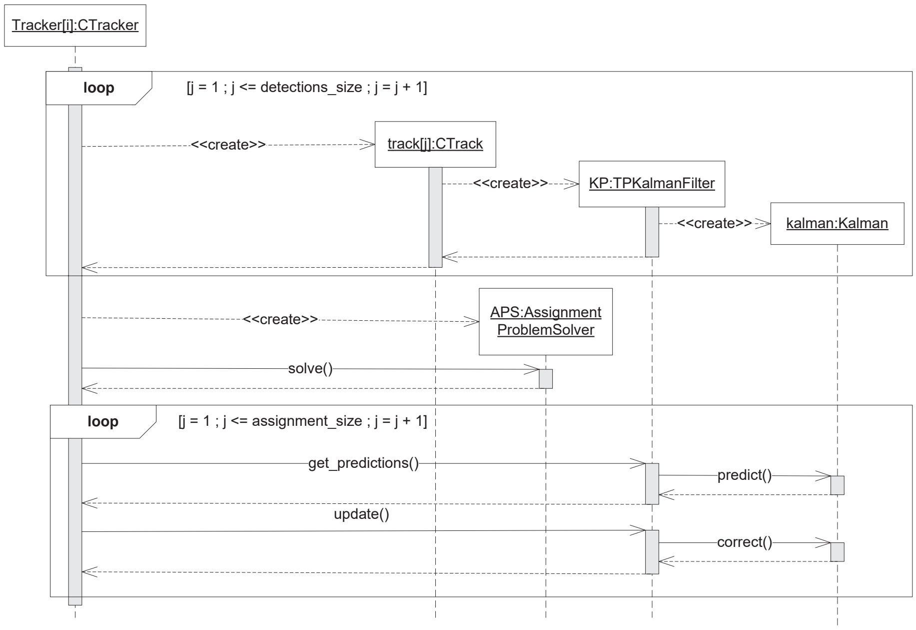


Fig. 8 Sequence diagram of function *update()*

and does not have high computational cost. Also are used a shadow removal algorithm using physics-based features because it provided good results with the BGS algorithm, and morphological operations to remove false foreground objects and to fill holes in the vehicles detected. At the third step the vehicles are detected using edge detection algorithms, and Kalman filters are used to do the tracking and the Hungarian algorithm for the assignment. The counting of the vehicles and their classification according to the area are done at the last step. The algorithm was implemented in a Raspberry Pi 2 SBC, with a clock frequency that was increased to 1000 MHz in order to reduce execution time, and it was running simultaneously in parallel to be able to work in real time. The video resolution was 640x360, being this the lower resolution in which significant differences were observed between the areas of the different vehicle types. The videos were recorded at 15 fps to reduce computational costs. The proposed trailer type, self-sustaining, counting and vehicle classification system is the basis for the development of ITS techniques in variable environments. The feasibility and applicability of the approach was demostrated in changing situations in both day and night. Thanks to the mobile system it will be possible to obtain statistics of the traffic flow in areas of difficult access in a faster way. The data obtained will be analyzed in order to offer potential solutions to traffic chaos, alternative routes, etc.

ACKNOWLEDGMENT

The authors would like to thank National University of Asunción (UNA) and National Council of Science and Technology (CONACYT) of Paraguay for their financial support with this project.

REFERENCES

[1] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, “Comparative study of background subtraction algorithms,” *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033 003–033 003, 2010.

[2] A. Sobral and A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos,” *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.

[3] A. Sanin, C. Sanderson, and B. C. Lovell, “Shadow detection: A survey and comparative evaluation of recent methods,” *Pattern recognition*, vol. 45, no. 4, pp. 1684–1695, 2012.

[4] Z. Chen, T. Ellis, S. Velastin *et al.*, “Vehicle detection, tracking and classification in urban traffic,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 951–956.

[5] N. C. Mithun, N. U. Rashid, and S. Rahman, “Detection and classification of vehicles from video using multiple time-spatial images,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1215–1225, 2012.

[6] L. Unzueta, M. Nieto, A. Cortés, J. Barandiaran, O. Otaegui, and P. Sánchez, “Adaptive multicue background subtraction for robust vehicle counting and classification,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 527–540, 2012.

[7] J.-C. Lai, S.-S. Huang, and C.-C. Tseng, “Image-based vehicle tracking and classification on the highway,” in *Green Circuits and Systems (ICGCS), 2010 International Conference on*. IEEE, 2010, pp. 666–670.

[8] A. Sánchez, E. Nunes, and A. Conci, “Using adaptive background subtraction into a multi-level model for traffic surveillance,” *Integrated Computer-Aided Engineering*, vol. 19, no. 3, pp. 239–256, 2012.

[9] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

[10] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.

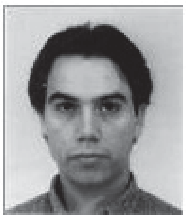
[11] J.-B. Huang and C.-S. Chen, “Moving cast shadow detection using physics-based features,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2310–2317.

[12] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[13] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.
[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[15] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

Derlis Gregor is Professor of Distributed Systems, Engineering Faculty of the National University of Asuncion (FIUNA), Paraguay. He received a Bachelor's Degree in Systems Analysis and the Computer Engineering from the American University, Asuncion, Paraguay, in 2007. He received a M.Sc. and a Ph.D. Degrees in Electronic, Signal Processing and Communications from the University of Seville, Spain, in 2009 and 2013, respectively. In 2014, he received an honorable mention at the National Science Award, Paraguay. He is a researcher categorized Level 1 by the National Council of Science and Technology (CONACYT), Paraguay. He is currently Head of the Laboratory of Distributed Systems (LSD), Engineering Faculty of the National University of Asuncion, Paraguay. His research focuses on the application of Intelligent Transportation Systems (ITS). Interoperability in Sensor Networks, Embedded Systems and Instrumentation Systems. Distributed Systems applied to Renewable Energy. Development and Management of Open Source software project for Embedded Systems.

Kevin Cikel received his bachelor degree in Electronic Engineering from the Engineering Faculty of the National University of Asuncion, Paraguay. He joined to the Laboratory of Distributed Systems (LSD), Engineering Faculty of the National University of Asuncion in 2015. His research focuses on the application of Intelligent Transportation Systems (ITS). Embedded Systems. Distributed Systems. Development and Management of Open Source software project for Embedded Systems.



Mario Arzamendia received his bachelor degree in Electrical Engineering from the University of Brasilia (Brazil) in 2002 and his master degree in Electronic Engineering from Mie University (Japan) in 2009. From 2009 until 2013 he worked as a project leader at the Automation and Control Innovation Center (CIAC) of the Itaipu Technological Park. In 2013 he joined the Faculty of Engineering of the National University of Asuncion as a researcher and since 2014 he is coordinator of the Department of Computer Science and responsible for the Laboratory of Distributed Systems. His research interests include embedded systems and wireless sensor networks.

Raúl Gregor was born in Asuncion, Paraguay, in 1979. He received the M.Sc. and Ph.D. degrees from the University of Seville, Spain, in 2006 and 2010 respectively. He joined Faculty of Engineering of the National University of Asuncion, Paraguay, in February 2009. Since 2012, he is the Head of the Department of Power and Control Systems, National University of Asuncion. Prof. Gregor received the Best Paper Award from the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS in 2009, and the Best Paper Award from the Institution of Engineering and Technology ELECTRIC POWER APPLICATIONS, in 2012.