

# Video Processing Techniques for Traffic Information Acquisition Using Uncontrolled Video Streams

Pedro F. Q. Loureiro, Rosaldo J. F. Rossetti, *Member, IEEE*, Rodrigo A. M. Braga

Department of Informatics Engineering  
Artificial Intelligence and Computer Science Lab  
Faculty of Engineering, University of Porto  
Rua Dr Roberto Frias S/N, 4200-465 Porto, Portugal  
{ei04062, rossetti, rodrigo.braga}@fe.up.pt

**Abstract**—This paper reports on the first steps taken in search of a solution that uses public video streams available on the Internet to address the increasing need for monitoring transportation networks with the intent of returning added value to the community, either by allowing a better understanding of the network and its needs or by feeding applications with real-time information for various purposes, such as simulation, decision-making support and updated route guidance. After the introduction of the field, we present our findings from a survey that briefly describes several works with related studies and explain the algorithms that can be adopted to get relevant information from video streams. This is followed by an analysis of the issues that may arise and the best ways to address them. Next it reports on the results achieved so far, draws some conclusions on what has been done and suggests the next steps of our research.

*Image processing and analysis; public video streams; traffic flow characterization; traffic monitoring*

## I. INTRODUCTION

In the last years, the growth of the number of vehicles in urban areas has led to an increase of the need of classifying traffic conditions in traffic networks. The information collected may have several purposes: control of vehicular networks, network management, planning, decision support, providing either simulators or route guidance applications with real-time data, improving the capacity of infrastructures to accommodate more cars everyday and enabling emergency vehicles to better serve the community.

The most usual characteristics that can be extracted are the number of vehicles on the road and their speed. With this information it is possible to estimate the traffic congestion. Some ways to collect these data which are currently in use include loop detectors, manual counting, radars and dedicated video cameras, to mention a few.

This paper, among all of the methodologies, suggests the use of public video streams as a source of information for this task. Although video cameras are already used, they often belong to a CCTV system or they are public but underused. Public video streams are observed by human controllers or used directly by the target population instead of having some automatic processing and providing the general population with treated and improved information.

The use of video cameras as main source of information is preferable to other alternatives as it does not require direct intervention to install magnetic loops on the streets, or any other type of expensive infrastructure. Indeed, having people manually counting vehicles passing through some section of a street is rather ineffective, and can only be made during experimentation periods. This approach is virtually a cost-free solution as it uses already installed infrastructures, such as public webcams and the Internet, and can be run 24/7 as it does not require human control or operation (although it can greatly benefit from the support of human operators). Additionally, it can be kept running after experimentation periods owing to its low operating cost. The implementation of the proposed system opens up the opportunity to fulfill the increasing need of cities to benchmark their infrastructures and transportation systems. This can be achieved by the creation of a daily source of information that can be used, for instance, to collect statistics and monitor performance as well as other needs of the infrastructure.

This work reports on our first steps towards the implementation of a framework for experimentation in order to give us an initial insight into how feasible the proposed platform will be. More specifically, our objectives are to carry out a preliminary study on the viability of implementing this solution and to identify related problems, techniques and available methodologies.

The remaining part of this paper is structured as follows. A literature review is presented in the next section, which includes both related work and common techniques in the study field. After that, there is an analysis of the challenges of public streams, as well as their problems and solutions. Following, there are sections dedicated to the preliminary work developed as well as to conclusions and future research.

## II. LITERATURE REVIEW

### A. Related Work

An information source for techniques of background subtraction applied to detection of moving vehicles and pedestrians, ranging from the most simplistic to the more complex algorithms is available in [1]. Aiming at optimization, [2] presents a methodology that improves the algorithms based on

adaptive background mixture models by using more intelligent updating equations which allows it to gain speed and more accuracy, as well as to adapt more effectively to changing environments. Additionally, the suggested technique is able to detect shadows.

One alternative is presented in [3], which proposes a new method to decompose the background and foreground in video frames. It is capable of detecting moving and static lines through the use of different filters and, at the same time, robust to cope with sudden illumination changes and computationally feasible to be implemented and used in real-time systems. Authors claim the results shown prove its effectiveness by decomposing background and foreground from video frames.

A novel method for detecting and segmenting foreground objects from video, with both stationary and moving background objects and subject to gradual or sudden changes is presented in [4]. According to the authors, *“the convergence of the learning process is proved and a formula to select a proper learning rate is also derived. Experiments have shown promising results in extracting foreground objects from many complex backgrounds including wavering tree branches, ickering screens and water surfaces, moving escalators, opening and closing doors, switching lights and shadows of moving objects.”*

A common problem in urban areas is addressed in [5], which proposes an enhanced version of a sigma-delta background estimation method, which is optimized for urban traffic scenes that are frequently affected by vehicles moving slowly or temporarily stopped. On the other hand, in [6] authors address the problem of refining moving objects by processing the background subtraction results. With this technique, it is possible to remove sudden illumination changes, local reflected regions and moving cast shadows. The results demonstrate that this solution is efficient for a wide range of different conditions that may occur.

In order to achieve the detection of vehicles under congested conditions, [7] suggests a Hidden Markov Model based algorithm to detect vehicles, which is capable of dealing with objects occlusion. It first extracts features from the images and then classifies them into one of three categories, namely road, head and body of a vehicle. Vehicles are detected by analyzing the sequence of categories found.

With a different approach, [8] introduces a vision-based vehicle detection method that considers the lighting context of the images and applies the best classifier algorithm to that situation. In its implementation, four categories of light were defined, namely daylight, low night, night and saturation. The results show considerable improvement in the detection performance with the use of a context adaptive scheme.

Methods for detecting and tracking vehicles, able to measure their speed at a distance of 70 to 100 meters (for incoming or rear viewed vehicles) are discussed in [9]. The detection uses a block-based algorithm and tracking is made with a variant of an extended Lucas Kanade template matching algorithm. This algorithm has detected vehicles with a speed accuracy of 2.3% for 95% of the vehicles, both in day and night time

sequences.

In a more particular but not less interesting application, [10] presents a system for automatic daytime road traffic control and monitoring system that retrieves traffic information such as mean speed, dimension and vehicles counting through computer vision approaches. It uses a frame differencing algorithm and texture information to extract moving objects from the scene and then removes shadows from the foreground objects with morphological operators. Objects are afterwards tracked using a Kalman filtering process and parameters such as position, dimensions, distance and speed are measured. Results from real outdoor videos show accuracy under daytime interurban traffic conditions. A different methodology for traffic flow characterization is described in [11]. It applies a multi-camera system, with omnidirectional and pan-tilt-zoom cameras combined. The latter is used to refine the information retrieved by the omnidirectional cameras.

A different approach to vehicle segmentation using a quad-tree segmentation is suggested in [12]. To accomplish the task of tracking vehicles, a scale invariant feature transform is used. It is able to extract parameters such as vehicle counts, speed and class. This approach has proved to be effective and robust, specially when the phenomenon of vehicle occlusion occurs, in the cases of a vehicle changing its lane and when the affine shape of a vehicle changes because of its movement. Also, targeted at heavily congested traffic situations, [13] proposes an algorithm that detects and tracks vehicle corners and then groups them into vehicles. The results show that this algorithm is effective under the assumed circumstances.

In this section, we have briefly presented some situations in which there was a need for processing videos to tackle particular problems. Most of the approaches were based on the difference between two images that may be the background estimation or the previous frame or rely on visual cues. The tasks performed involve estimating features such as speed or ambient conditions in order to improve effectiveness through the use of specialized algorithms. In next section, possibilities to approach these situations are analyzed as well.

## B. Related Techniques

Shadow detection and removal, as well as vehicle occlusion are recurrent difficulties. Motion detection is a very important task as it is one major feature of traffic monitoring applications: cars are moving. There are several techniques to segment vehicles from the background of a scene picture or of a video frame. The most widespread techniques involve the calculation of the optical flow. In optical flow, we are interested in finding vector fields that describe the way that the image is changing. In other words, we try to detect where certain parts of the image have moved to in the next frame. Such techniques usually have two steps: i) to detect feature points, and then ii) to track them over the frames. There are several ways of achieving this, some of which are described in [14] and [15].

Feature points are special points that are distinguishable from their neighbors. Each algorithm may use different strategies to detect and track these points. One alternative is



(a) Initial frame

(b) Second frame



(c) Highlighted square comparison

Fig. 1. Aperture problem example

calculating the difference between two consecutive frames. Pixels for which difference is bigger than a threshold are considered to be moving. However, this technique does not detect the direction of movement and can only be used for simpler applications. Additionally, this algorithm suffers from the aperture problem. This problem happens when a surface with a smooth color moves. As the color is similar in all parts of the surface, movement may not be detected in areas of the captured image where the surface was already present because the difference is not sufficient to consider it a motion and thus, it is interpreted as image noise or small variations in illumination. This problem is demonstrated in Fig. 1. It has two consecutive frames with the same region highlighted. In Fig. 1c, the same region in the two frames is compared and augmented. Although they display a different part of a moving vehicle, focusing on smaller portions only is not enough to perceive any difference.

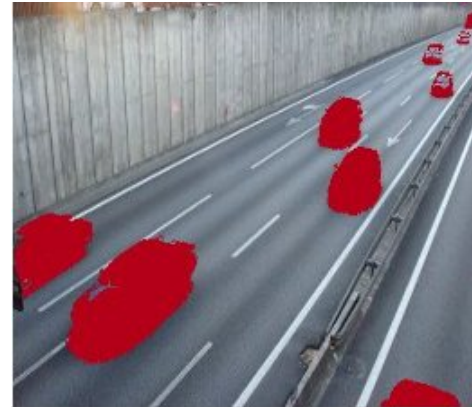
A different approach is used by background subtraction methods, which are also able to detect moving objects. One important premise in the method is that the background will have small changes over time and foreground objects such as cars are moving objects which will contrast with the background. Therefore, if foreground objects are detected it is safe to consider them as moving objects. Ideally, for traffic monitoring use, this background image would be the same frame without moving objects, showing the scene as if there were no cars or other moving objects on it. There are some alternatives to accomplish this.

The easiest method is choosing a fixed image that is the background image. After subtracting one image to another, those pixels whose difference is bigger than a threshold are considered to be moving pixels. This is usually too simplistic as changing illumination conditions, weather, noise or small



(a) Input image

(b) Background Estimation



(c) Foreground mask (in red) applied to input

Fig. 2. Background estimation example

movements of the camera may cause big differences when there are no moving objects in those regions.

Better options involve making a background estimation image, based on the frames captured by the camera. There are several methods to estimate the background and foreground of a sequence of frames, of which [16] presents an interesting overview. The main difficulties of these algorithms happen in situations with very slow traffic movement or temporarily stopped vehicles as they start blending with the conceptual background. A solution to this problem is suggested in [5], as well as is an alternative proposed in [3] to use time-varying background and foreground intrinsic images. Both studies achieved very good results in terms of successfully segmenting background and foreground. Fig. 2 shows the results of a background estimation algorithm after running for around 30 seconds. It has a background estimation which is very close to the real background and was able to successfully identify moving objects without any extra processing or calculation.

Vehicle segmentation is probably the most frequent type of information that is extracted from traffic surveillance videos. Its objective is to distinguish between vehicles from the background and also vehicles from each other. One common method to detect vehicles in a video sequence uses one of the previously analyzed motion detection or background estimation to detect the cars. A solution that compares the current frame with the background estimation using a quad-tree decomposition to find the pixels where cars are found is presented in [12]. As stated in its experimental results,

occluding vehicles may be grouped together as one object under heavy congested traffic conditions. A Hidden Markov Model is used in [7] to detect cars under congested conditions, where occlusion is frequent. A different approach, presented in [13], does not require the background estimation, but instead it detects information about the corners of the vehicles. Both [9] and [17] present a good survey of other vehicle detection algorithms, in which the latter proposes a way to combine several of these methods.

Another common task is shadow removal, with several benefits. In fact, the shadow of a moving object is also moving and hence could be considered a moving object. If the shadow of a car is considered together with the car, it may seem that the car is bigger than it really is. Another situation to bear in mind is when the shadow of a car connects to other cars, causing not optimized algorithm to merge both cars into one single object. A solution to remove shadows from images is then presented in [10], which uses top-hat transformations and morphological operators. In [6], a solution to remove shadows based on a single Gaussian shadow model is presented, whereas [18] describes other alternatives with references and further information about them.

Estimating the speed of vehicles is an important task because it can be used as a source of information for most of the traffic monitoring applications. This task is challenging owing to the difficulty in acquiring, lack of precision of means used and highly sensible data. Not even humans, observing attentively videos with good quality, can precisely tell which pixels belong to one car or another. Also, blur from cars moving at high speeds makes this even more difficult. Indeed, the exact position of the camera is not known (and cannot be calculated very precisely) besides vibrations and movement of the camera that cause unpredictable variations. All these difficulties together cause enormous noise when geometrical information that relies on the position of the camera and observed cars is required. This happens because the applied methods are highly sensible. In other words, small changes in one variable may induce into big differences in the final estimated velocity. In [9], authors estimate speed based on the point of contact of a car with the road, whereas [10] uses a geometrical equation to accomplish that.

Tracking is also one important application, being vehicle tracking the most common type. Nonetheless, it is also possible to track other features of an image. For a comprehensive discussion on tracking algorithms, the interested reader is referred to [12], which also suggests a methodology to track vehicles using a scale invariant feature transform. This algorithm, applied to each vehicle, will describe its features such as pixel values, key point locations and orientations into a 128 dimensional vector, which can be tracked in the following frames. In [13], instead, the suggested technique detects corners of vehicles, tracks them using Kalman filtering and groups the corners into vehicles.

### III. CHALLENGES OF PUBLIC VIDEO STREAMS

#### A. Problems Overview

One group of problems of special interest here is the direct outcome of the lack of control over the cameras. The placement and direction of the camera may affect in several ways how information is displayed. Most applications need to use several cameras so similar information may be displayed in different ways. One example would be cars moving from bottom to top in one camera and another camera in a similar situation showing cars moving from left to right. Another frequent problem is the low quality of the stream available and sometimes the frame rate. Although some algorithms can tolerate lower frame rates, that is generally not true to low quality images. One unique problem that arises from the use of external video sources is that some of them aggregate several cameras into one single feed, alternating the camera that is shown with a given frequency. Some issues commonly challenging in analyzing video streams include the fact that the distance of the camera to the road changes the apparent size of cars, multi-lane routes can support higher number of cars without being over-saturated, two-way or one-way streets and pedestrians walking on the sidewalk may introduce errors in algorithms unless they distinguish among people, cars and other moving objects, to mention a few.

Besides the above mentioned issues, one ought to consider another set of difficulties that are partially related to the camera's positioning but that even in ideal conditions cannot be fully solved. Examples of these include occlusion, the apparent change in size and speed of moving objects as they move away from (or into) the camera, and so forth. Even the best cameras available on the market produce images with noise that may cause problems to the most sensible algorithms.

However, the most challenging problems exist whether there is control over the video sources or not. Weather has a major role in these problems. The same scene has enormous differences if the weather is clean, rainy, foggy or snowy. This way, an algorithm that performs well under clean weather may be not acceptable under foggy or rainy conditions. Another difficulty is the ambient light. Indeed, the light in outdoor scenes changes dramatically throughout the day, specially in places outside urban areas where there is no need to illuminate streets at night. This brings about two problems. First, algorithms will work under totally different light conditions – literally, from night to day. Second, throughout the day the luminance is not constant, which affects some algorithms that rely on estimations of how the scene should be in a predetermined situation (e.g. without any cars). As the lightning conditions change, the estimation must be updated several times within a day.

#### B. Camera Related Issues

Camera tilt is a very important factor that must be taken into consideration as well. Tilting a camera is related to the angle that is made with the horizontal plan. In its full extent, camera tilt can range from horizontal to vertical pointing

cameras. The latter can be found in satellite views, very tall buildings, bridges or tunnels, if they point directly downwards. The main advantage of these cameras is the removal of vehicle occlusion problem but it has a drawback due to perspective. Thus, it can only analyze a much smaller area. Unless the camera is at some very high point (satellites or very tall buildings), the observable area may be so small that it is not able to properly track moving objects because they are only seen in a small number of frames before they move out of the observable area. On the other hand, cameras pointing (closer) to the horizon have a great observable area but it may be so big that it starts having some extra difficulties caused by the depth. Far away, objects and distances seem smaller than if they were closer to the camera. This must be taken into consideration in algorithms that try to estimate distances, speed or track object properties. Another problem is vehicle occlusion, which happens specially under congested traffic conditions as vehicles tend to move closer to each other, shortening the headway between them.

To address the issue of low video quality and/or frame rate, [19] describes and compares several filters and preprocessing techniques that can be used to improve the quality of the video. Three of the simpler alternatives have the advantage of being frequently used and therefore implemented in the majority of frameworks, namely the median, the average and the bilateral filters. The first two will give a pixel the value of the median (or average) of its surrounding pixels. This action smooths the image but also makes it more blurred. Bilateral filtering is a filter that tries to reduce this effect specially on the borders of objects, smoothing the color of objects but not their edges.

When similar information is displayed with different appearance, some algorithms may suffer a loss of effectiveness in some of the possible appearances. However, it is usually possible to tune one method to return better results for a particular situation, while making the other situations worse. The best way to overcome this problem is developing parameterizable algorithms. The most frequent parameters should be camera tilt, camera distance to the route, amount of light (or hour of the day), number of lanes, expected value for some variables, and so forth. Each algorithm may have a set of particularly interesting parameters so it is very important to understand the relation between the different scenarios and the corresponding optimization of the algorithm, and then model it through a set of parameters. This way, when a new environment is added to the system, instead of manually configuring all the variables, a smaller number of parameters might be given and the variables might be calculated based on them. In the case of more complex relations, artificial intelligence (AI) methods can be used to learn from examples given by the authors and help determine better values for other environments. The AI community is especially interested in further analyzing this kind of problems.

To tackle the problem of having several cameras in one single feed (or one camera that is not static and therefore can provide images from different places), one possible solution is to track the difference of consecutive frames. When this

difference is bigger than a threshold it is assumed that there has been a change in the camera's source (or direction), meaning that the new images belong to a different situation. Now, the observer must identify what the new scene is and which camera (or direction) the images belong to. Again, AI-based approaches such as pattern recognition techniques can do the task with sample images previously provided. There are still two remarks about this issue. First, one feed can only provide a small number of different scenarios so the algorithm should only try to distinguish among the possible scenarios for that feed and not all of the scenarios of the system. Second, in the case of cameras with some degrees of freedom, allowing them to point at several locations, it is also possible and very likely that a new scenario is detected while moving to a new position. If this happens, the frame should be ignored as there is no clear image to be analyzed. It could be detected by a low fitness of that frame to any of the possible scenarios of that feed, for instance.

### *C. Domain Intrinsic Difficulties*

Some of the difficulties exist regardless of any factors related to the camera or its operation. These difficulties are generally more studied due to their universal existence. Nearly every study about traffic monitoring using video cameras is affected by them.

Vehicle occlusion, for instance, is a problem that occurs when one car overlaps another from the perspective of the camera. This does not depend on any other factor but vehicles interacting throughout a normal daily traffic profile. It is rather frequent to happen to cameras which are aimed at some distant point (near the horizon), but it is also possible to happen to cameras seeing the route not directly from above it but laterally. Occlusion affects algorithms that try to use visual cues like the shape, contours or edges of objects because a partially occluded object will not look like a normal object and one perceived object may be in fact the junction of two self occluding objects. Fig. 3 illustrates this problem showing a big truck overlapping several cars. However, even small and sports cars can partially occlude bigger vans at long distances.

Weather is probably the biggest challenge to solve in this area. There are mainly four types of weather conditions that may happen, namely rain, snow, fog or clear weather. When referring clear weather, the most correct definition would be non rainy, snowy or foggy. This is because the difference between clear weather and clouded (but not raining or snowing) is negligible although some problems might be created owing to shadows and light changes, which are discussed in other parts of this paper. One common feature to all of those non clear weather situations is the loss of visibility. This is specially true for fog, as it is its main effect together with partial loss of color. The difficulties created under rainy conditions come from the reflections of car lights on the surface of the road that, when cars are facing the camera, may turn the images into a white stain. Snowy conditions turn a big part of the image into white, including moving objects (which may have layers of snow on top of them).



Fig. 3. Occlusion of several vehicles

This makes it more arduous to distinguish objects from the background, distinguish objects from each other and detect moving objects. Another factor shared by rainy and snowy conditions is the possibility of having drops of rain or flakes of snow on the camera lens, blocking part of the image and making it impossible to be analyzed. The best way to solve the problems of weather is using algorithms specialized in those particular conditions as there are not any universal solutions suitable to cope with all conditions. However, it is important not to ignore those conditions in real-life applications and experiment them under all possible conditions as unexpected results may occur otherwise.

As for light conditions throughout the day, there are basically two types of problems that may happen. The first one is the big difference that happens from day to night. The second one is small differences that happen throughout the day. This affects the algorithms that assume that light is exactly the same in small intervals of time and use an auxiliary, fixed image, usually a background, to compare with the new image. Although light may seem constant, small differences must happen in order to have a big difference after 12 hours. This is however easily solved by applying threshold techniques (e.g., to consider that one pixel has changed since the last frame, its difference must be bigger than a fixed value) or by updating the background image (or any resource that expects fixed amount of light) when considerable differences of light between current frame and auxiliary image are detected. To address the issue of day-night conditions, although it might be possible to use parametrized algorithms (specially where there is good artificial illumination), the use of algorithms specialized in night conditions may be the only solution. Sometimes, the specialized algorithm may use a totally different approach. For example, to detect cars during the day, it is possible to detect them by their shape, whereas detecting car lights is usually a better option at night.

## IV. PRELIMINARY DEVELOPMENT

### A. Methodology

To devise a typical application in this area involves three steps: detect moving objects, segment them and track them. In order to facilitate the development of the application, steps must be taken one at a time. This means that the easiest task is implemented first, in a simple context. Simple contexts include, among others, one static image produced to test a specific algorithm, static images taken from videos or photographs with good conditions like color, contrast or scenario, a sequence of created images or video, and so forth. When one task is successfully performed under simple circumstances, it can be tested under real videos, starting by the videos with better quality or scenarios.

In practice, this is not followed all the time for practical reasons. Usually it is required that a set of tasks be executed at a time, not only a single one. Thus, as soon as the first task produces acceptable results under acceptable conditions, the next step is started to be implemented. When all of the steps are implemented, another cycle takes place to further improve them, applying them in more realistic conditions.

Additionally, it is important to use a framework, library or some toolkit to help the development of the application, avoiding to recode known algorithms and simplifying the design and implementation of the graphical user interface (GUI). Another important investment is to use an auxiliary tool to help the development and debugging of the application.

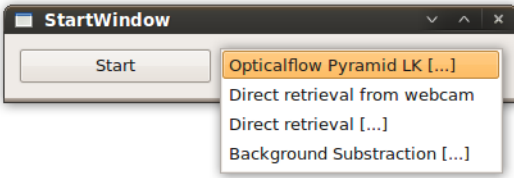
### B. Prototypical Studies

OpenCV has been chosen as the underlying application programming interface (API) for the development of our platform, basically due to the fact that it is a complete, widespread and popular library with a strong community support. One additional tool was also implemented to allow easy debugging and on-the-fly configuration of algorithms without requiring recompilation of the program. Fig. 4 shows the GUI of the testing framework developed.

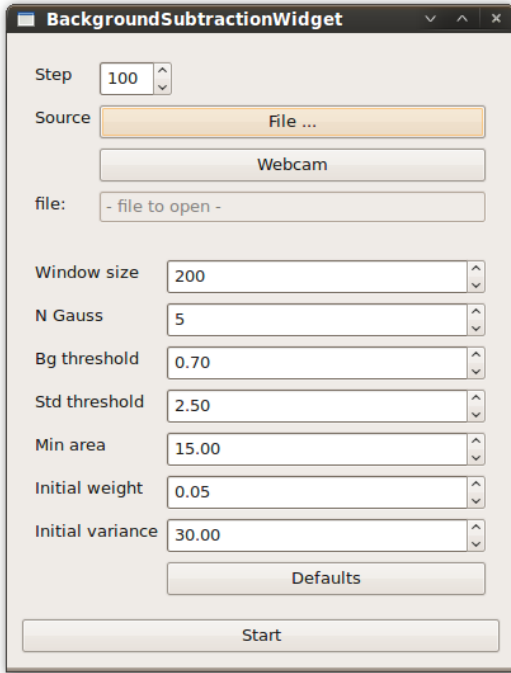
Despite being our original interest the characterization of traffic flow on aggregate basis, first steps taken toward the implementation of our application have shown some good potential for vehicle detection as well. Both optical flow, with or without pyramidal segmentation, and background subtraction using the Gaussian background-foreground estimation model demonstrated promising results. However, these results still require improvements that might likely be accomplished by morphological operators or by improving the quality of the video. The use of contour and blob detection algorithms achieved good results that however declined with the increase of the density of vehicles, when occlusion begins to exist.

## V. CONCLUSIONS

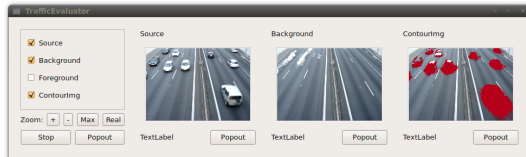
In this study we have successfully identified an area of application for public video streams, which will produce great benefits for the community in general and for the industry as well. We have carried out a survey that can be used as a starting point for other studies in the field or related



(a) Selection of Algorithm



(b) Configuration Window



(c) Results Window

Fig. 4. The testing framework developed

areas. By identifying the relevant issues in using public video streams, we were able to devise a methodological approach to start digging a bit deeper into the development of a platform to aid traffic flow characterization. Nonetheless, a similar methodological approach may be applied to other analogous problems.

After the initial survey which gave us a general insight into this area and its main issues, a testing application has been designed and iteratively developed. The next steps include the improvement of the results achieved so far. This will be performed in two basic ways. Firstly, we intend to test with different parameter values, tuning the configuration of the implemented algorithms. Secondly, we plan to test with alternative algorithms and extend the ones already developed.

For the latter case, AI-based techniques will play a major role.

Additionally to analyzing traffic on an aggregate basis, individual vehicle detection and tracking are also features to be developed. Inarguably, this will pose great challenges and will bring about tricky issues to overcome when public video streams are to be used. Finally, a comprehensive and thorough calibration and validation methodology must be devised so as derived traffic flow information can be used and integrated into other decision support systems.

#### ACKNOWLEDGMENT

Authors gratefully acknowledge the support of NDrive, Navigation Systems.

#### REFERENCES

- [1] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Visual Communications and Image Processing*, vol. 5308, 2004, pp. 881–892.
- [2] P. Kaewtrakulpong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, vol. 5308, 2001.
- [3] F. Porikli, "Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images," in *Proc. of IEEE Motion Multi-Workshop*, 2005.
- [4] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. 11th ACM international conference on Multimedia*, 2003, pp. 2–10.
- [5] M. Vargas, S. Toral, F. Barrero, and J. Milla, "An enhanced background estimation algorithm for vehicle detection in urban traffic video," in *IEEE Conf. on Intelligent Transportation Systems*, 2008, pp. 784–790.
- [6] K. Wang, Q. Yao, X. Qiao, S. Tang, and F.-Y. Wang, "Moving object refining in traffic monitoring applications," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 540–545.
- [7] M. Yin, H. Zhang, H. Meng, and X. Wang, "An HMM-based algorithm for vehicle detection in congested traffic situations," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 736–741.
- [8] D. Acunzo, Y. Zhu, B. Xie, and G. Baratoff, "Context-adaptive approach for vehicle detection under varying lighting conditions," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 654–660.
- [9] B. Alefs and D. Schreiber, "Accurate speed measurement from vehicle trajectories using adaboost detection and robust template tracking," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 405–412.
- [10] P. F. Alcantarilla, M. A. Sotelo, and L. M. Bergasa, "Automatic daytime road traffic control and monitoring system," in *IEEE Conf. on Intelligent Transportation Systems*, 2008, pp. 944–949.
- [11] R. Khoshabeh, T. Gandhi, and M. M. Trivedi, "Multi-camera based traffic flow characterization & classification," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 259–264.
- [12] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang, "Multiple vehicles detection and tracking based on scale-invariant feature transform," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 528–533.
- [13] Z. Yang, H. Meng, Y. Wei, H. Zhang, and X. Wang, "Tracking ground vehicles in heavy-traffic video by grouping tracks of vehicle corners," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 396–399.
- [14] S. M. Smith, "Reviews of optic flow, motion segmentation, edge finding and corner finding," Oxford University, Tech. Rep. TR97SMS1, 1997.
- [15] S. Beauchemin and J. Barron, "The computation of optical flow," 1995.
- [16] M. Piccardi, "Background subtraction techniques: a review," in *IEEE Conf. on Systems, Man and Cybernetics*, vol. 4, 2004.
- [17] Y.-M. Chan, S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, "Vehicle detection under various lighting conditions by incorporating particle filter," in *IEEE Conf. on Intelligent Transportation Systems*, 2007, pp. 534–539.
- [18] F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2003.
- [19] A. C. Bovik, *Handbook of Image and Video Processing*, 2nd ed. Academic Press, Inc., 2005.