Post-processing of Videos for Vehicle Counting

Abel Ricardo Marcão Ribeiro and Prof. Paulo Lobato Correia

Abstract—Automatic counts of vehicles are a important component of Intelligent Transportation Systems (ITS) applications, which aim to provide services related to distinct modes of transportation and traffic management to allow the users to be better informed and make better use of the transport network. The goal of the ITS is contribute to a smarter use of the transportation networks allowing a better management of time and resources.

This paper develops a Video-based Traffic Management System to count vehicles acquired from Traffic Surveillance Video (TSV) streams available on the Internet. The system obtains Temporal-Spatial Images (TSI) from the TSV streams and computes a combination of features to acquire the vehicles on the videos. The vehicle counts can be provided to traffic guidance applications, allowing better route selection and thus improved decision-making.

The developed system has flexibility to allow the selection of the road lanes which should be counted; it is robust, adapting itself to different operational conditions on the scene being captured, such as night and day time and inclement weather conditions, without any human operator interference; it has a fast processing speed which allows real-time counts of the vehicles and an occlusion detection method to provide improved counts.

Index Terms—Traffic Management System, Temporal-Spatial Image, Traffic Surveillance Video, Virtual Detection Line, Occlusion Detection

I. INTRODUCTION

The increasing number of circulating vehicles has called for the development of more efficient systems to classify and monitor road traffic. The information acquired by these methods is used to plan and to manage the road network, to allow monitoring of the vehicles on the network and to provide real-time data to guide the users of the network. These systems enhance the capacity of the road network infrastructure, improving drivers’ safety and traffic management efficiency.

Traffic management with real-time data gains special importance in densely populated areas, due to the high amount of vehicles circulating in them, as it provides the means to better assess traffic conditions and thus be able to divert it in case of unpredictable events such as car accidents, which may cause long lines of vehicles and traffic jams, eventually leading to more accidents due to the confusion originating from these events. Hence, the importance of Intelligent Transport Systems (ITSs) development, which should be fast enough to reach and divert the traffic in such events.

The European Commission [1] defines the ITS as advanced applications which aim to provide services related to distinct modes of transportation and traffic management to allow the users to be better informed and make better use of the transport network.

The use of video cameras for monitoring purposes is becoming more popular overtime. Hence, a video-based road traffic monitoring system is proposed in this paper. This system provides a traffic assessment through vehicle counting using video and image processing methods. Any device such as a GPS or a cell phone could receive the result of the processed traffic information and use it to decide the best route to take and thus avoid traffic congested paths.

The system was developed to be able to work with low resolution videos, which are the type of videos typically provided online for the majority of the traffic surveillance cameras. The proposal includes an occlusion detection method, ensuring the system robustness, allowing to achieve high accuracies in all tested operational conditions.

The remainder of the paper is organized as follows: Section II reviews the state of the art on traffic management systems, with a special focus on video processing techniques. Section III presents the proposed system, detailing the used methods. Experimental results are presented in Section IV. Section V identifies the strengths and weaknesses of the proposed technique, providing some suggestions for future work.

II. STATE OF THE ART

Traffic management is a field primarily dedicated to improve the flow of vehicles and improve road’s safety. It involves traffic monitoring and classification by local or national roadway authorities to manage traffic flows and provide advice concerning traffic congestion.

The video-based classification methods usually start by a foreground estimation of video or image, by removing the background scene and thus highlight the moving objects that are considered candidate vehicles. In this phase a background model is typically used. The next step receives the mask of the foreground estimation, to detect the moving objects. A detection algorithm is used to identify the objects, generating an object list, and extract the relevant features from them. In a third phase, the objects in this list are classified has vehicles or not, based on a set of defined characteristics that should distinguish the vehicles from any other possible moving object thus avoiding false positives. As an output of this phase, we should have a list of vehicle allowing their counting and, depending on the algorithms used, their tracking. This generic architecture is illustrated in Fig. 1.

Abel Ricardo Marcão Ribeiro and Paulo Lobato Correia are with Instituto de Telecomunicações – Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001 Lisbon, Portugal (e-mail: hds@lx.it.pt, plc@lx.it.pt).

The authors acknowledge the support from Instituto de Telecomunicações.
Fig. 1: Block diagram of a traffic management system

A. Challenges

The detection of vehicles in the foreground is crucial for video-based traffic management systems. The use of public, uncontrolled, TSV streams is challenging. Loureiro et al. [2] present a comprehensive inventory of the problems arising from the use of such TSV streams for traffic monitoring. These range from undesired camera tilts, to overall low-quality videos, shadows and vehicle occlusions caused by low-angle image acquisition. Possible solutions to address the above challenges [2] are listed below.

**Shadows**- The shadows of moving objects need a careful consideration in traffic management systems. The detection of moving shadows affects the efficiency of the system as the shadow of a moving vehicle is also moving and could therefore be detected as a different vehicle. When the shadow of a vehicle connects to other vehicles, the detection algorithm may merge them into a single object. Problems caused by shadows are detailed on [3-7].

The methods used in the literature to deal with shadows are divided in two types, property-based and model-based. On one hand, property-based methods use extracted features from the objects, such as geometry, color or brightness, to identify the shadow regions of the objects. On the other hand, model-based methods use a priori knowledge of the scene geometry, the foreground objects or the light sources to detect the presence of moving shadows on the objects [8].

**Inlement Weather**- Distinct types of weather provide different challenges to a traffic monitoring system. Rainy weather may cause puddles on the roads which cause reflection in the video’s image and may cause the detection of false vehicles. The visibility of the traffic from the cameras tends to be reduced, being specially noted on foggy weather. This reduced visibility is of special importance, because it hinders the vehicle detection, due to the reduction of the pixels intensities which cause a lower variation between them and reduce the efficiency of the object detection. Once more, the related literature doesn’t provide significant specialized research in these situations [9]. To mitigate the lower efficiency caused by lower variations between pixels, the detection methods tend to be adjusted with higher sensitivity to pixels changes. These changes usually are made by lowering thresholds or reducing the size of smoothing filters used to reduce the noise, such as the Gaussian Filter.

**Vehicle Occlusion**- It occurs when the view of an object is partially or completely obstructed by another. In the cameras used in traffic management systems, this is usually caused by the high traffic density on the road or by a poor angle of the camera placement. As a result, vehicles might be missed, reducing the overall system performance. To avoid these situations in a traffic management system, the cameras should be placed in high poles over the road, which provide a better view angle of the road and the vehicles [9].

B. Motion-estimation-based vehicle detection methods

Vehicles are usually detected from TSV streams by finding objects with signification motion. There are several researched methods, which include Gaussian scale mixture model method [10], the interframe difference method [11], object-based segmentation [12] and background subtraction method [13]-[15].

The detection method used on this paper is based on Virtual Detection Lines (VDLs) and a modified background estimation method developed in this paper. In this method, Temporal-Spatial Images (TSIs) are generated using the luminance values of pixels of the moving objects which cross the VDLs [16]-[19]. Afterwards, the moving objects are detected when they cross these VDLs by applying the background subtraction to the TSIs. This background subtraction is acquired using the background estimation proposed in this paper. VDLs and a TSI with the detected moving objects are illustrated in Fig. 2, respectively on the left and right.

**Fig. 2: Examples of VDLs and TSI**

**III. PROPOSED SYSTEM**

The system proposed in this paper is a Video-based Traffic Management System, which uses an image processing algorithm to estimate the number of vehicles from uncontrolled traffic monitoring video cameras. The system has to be robust enough to handle distinct operational conditions, such as night time or inclement weather, and also be able to minimize the number of counting errors due to occlusions.

The TSV streams tend to have low resolutions, usually around 200x200 pixels or less, and the acquiring cameras are installed far from the road lanes, thus the visualized vehicles mostly appear as small objects with a minimum amount of detail. Therefore, the methods used on this work had to rely on object features such as area, position and moving direction to decide if it should be considered as a possible vehicle.

A. System implementation

The system thus follows a sequence of steps to acquire the features that allow it to obtain a vehicles list.

**Initialization** – System initialization consists of using the application’s GUI to draw the VDLs on the TSV’s initial frame, from which the ST images will be automatically created. The
user can draw an unlimited number of VDLs, in order to analyze each road lane individually or all the lanes together. There are no limitations of direction, angle or length. In order to obtain the best results, the VDLs should be drawn perpendicularly to the road lanes.

**TSI Generation** – The TSIs of the VDLs drawn on the previous step are generated individually. The length of the TSI depends on the number of frame present in the video under analysis and the width of the TSI depends of the length of the corresponding VDL, as illustrated in Fig. 3.

![Fig. 3: VDL and corresponding TSI](image)

**Background Estimation & Subtraction** – A new method was developed to find the best estimation for the background of the TSI. Traditionally, the background estimation of a TSI is performed by searching the column of the TSI with lower variance and using this column as the background subtraction to the whole TSI. The rationale is that the background of the TSI is usually the road pavement and its pixels should have nearly constant intensity and thus this lower variance approach can find it. This approach works correctly when the road present in the TSI is just the asphalt. However, when there are road surface markings this rationale does not hold true most of the times. If, for instance, one of these road surface markings becomes hidden by a dark vehicle, using the lower variance approach, one of the columns of the TSIs where the marking is hidden will be chosen as the background instead of a column where the pavement markings are present. This will provide the wrong background as illustrated in Fig. 4 on the right.

The approach proposed in this paper uses a histogram of the TSI, to give score to each pixel’s intensities. The score is based on the frequency of that intensity on the TSI. More frequent intensities have higher scores. Hence, all the TSI pixels are given a score following the equation (1).

\[ V_I(x,y) = h(I(x,y)) \]  

Where \( I(x,y) \) is the matrix of the TSI, \( V_I(x,y) \) is the score matrix of all the pixels and \( h \) is a function based on the histogram that gives higher scores to the intensity values with higher frequency.

In order to acquire the sum of the score on each column the sum of each column of \( V_I(x,y) \) is computed, which is defined by equation (2).

\[ v_x = \sum_{i=1}^{M} V_I(x,i) \]

Where \( v_x \) is a vector with the scores of all the columns. Finally, to obtain the column of \( V_I \) with highest score, it can be computed by equation (3).

\[ v_x^* = \max\{v_x\}, \forall x \in \mathbb{Z} \]  

The column with index \( s \) has the highest score and thus is the column of the TSI which has the pixels that are more frequent in the whole TSI. Hence, it has the highest probability of being a good representation of the background of the VDL. The difference between both approaches in a road with surface markings is illustrated on Fig. 4.

![Fig. 4: Left: Approach developed in this paper; Right: Lower variance approach.](image)

**Edge detection** – The edge detection tries to find the moving objects on the TSI. It works on the modified TSI, which is the result of the background subtraction of the previous step. This modified TSI has the moving foreground objects which includes the moving vehicles on the road.

The Canny edge detector was chosen to extract the edges of the foreground-only TSI due to its resilience to image blur, arising from fog and other adverse weather conditions, and due to the high correlation between the contrast and the edges in an image.

The Canny edge detector contains two threshold parameters, which have to be adapted according to the illumination present on the video. The ratio 1:3 between the low threshold and the high threshold presented the best results for the vehicle detection.

The value of threshold varies not only accordingly to the illumination present on the road pavement, measured by the average pixel intensity of TSI, but also accordingly to the overall illumination of the video, measured by the average pixels of a sample image taken from the video. This led to the development of the 3 modes:

- the Day Mode (DM);
- the Night Front Side Mode (NFSM);
- the Night Back Side Mode (NBSM).

Each of these modes has its own settings for the Canny low threshold. The day mode is used on bright videos, usually during the day. NFSM and NBSM are used on dark videos, usually at night. NFSM is used when the vehicles are moving towards the position of the camera, otherwise the NBSM is used. The variation of the Canny low threshold on the 3 modes is illustrated in Fig. 5 and Fig. 6.
The NFSM has a constant value of this threshold because the detection of the vehicle is made slightly differently on this mode. In this case, the TSI take an additional modification before applying the Canny edge detector. Otsu’s method is applied to the TSI in this mode in order to highlight the headlights of the vehicle and ignore everything else. The rationale is that due to the headlights higher intensity, they are easier to detect than the body of the vehicle and thus they are used for the vehicle detection. Therefore, the Canny edge detector can have a constant and high threshold independently of the average pixel intensity of the TSI.

**False Vehicle Elimination** – There are situations where undesired objects are identified causing false vehicle detections. These spurious objects usually appear due to noise or double detections of the same vehicle. In order to eliminate these undesired objects, a verification of the corresponding features, namely their area, width and height, allow deciding whether to keep them. The method proposed and developed in this paper for False Vehicle Elimination (FVE) is composed of three processes for vehicle validation, as described in the following paragraphs.

The objective of the first proposed process is to eliminate the objects which have small and unusual dimensions, namely a small width and an extremely long height or a small height and an extremely long width. These objects are noise that reduces the vehicle counting accuracy. To do so, the width and height of the objects are analyzed. When equation (4) is satisfied, the object is eliminated. Otherwise the object is kept.

\[
\left( \text{height}_i < \delta_h \land \text{width}_i < \frac{\text{height}_i}{\alpha} \right) \lor \left( \text{width}_i < \delta_w \land \text{height}_i < \frac{\text{width}_i}{\alpha} \right) \quad (4)
\]

Where: \( \text{width}_i \) is the width of the object \( i \) in pixels; \( \text{height}_i \) is the height of object \( i \) in pixels; \( \delta_h \) and \( \delta_w \) are values for minimum height and width of the object in the interval and \( \alpha \) is a ratio between the width and the height of the object.

The second developed process is used to verify the objects’ area using the inequation (5). Its purpose is to reject small objects, usually noise, as vehicles. When the inequation is satisfied, the object is rejected. Otherwise the object is not rejected.

\[
\text{area}_{\text{object } i} < \alpha \cdot \text{area}_{\text{average}}, \quad \forall i \in X
\]

Where: \( X \) is the set of objects detected; \( \text{area}_{\text{object } i} \) is the area of the object \( i \); \( \text{area}_{\text{average}} \) is the computed average area of the objects detected; \( \alpha \) is an arbitrary value chosen by the user.

The third verification introduced in the FVE developed in this paper searches for the duplication of objects for the same vehicle, due to multiple detections of the same vehicle. The multiple detections can usually be removed by eliminating the detected objects which are at least partially overlapped. The strategy used to tackle this is based on the comparison of the overlapping area, with the total area of each of the individual rectangles. When this overlapping area is larger than a given threshold, the rectangle is rejected. However, both rectangles cannot be simultaneously rejected; hence when a rectangle is rejected the one with a bigger area is always kept.

The overlapping area is computed using the equations (6), (7) and (8).

\[
\text{width} = \min\{x_a + \text{width}_a, x_b + \text{width}_b\} - \max\{x_a, x_b\} \quad (6)
\]

\[
\text{height} = \min\{y_a + \text{height}_a, y_b + \text{height}_b\} - \max\{y_a, y_b\} \quad (7)
\]

\[
\text{area}_s = \max(0, \text{width}) \cdot \max(0, \text{height}) \quad (8)
\]

Where the rectangle \( a \) and \( b \) are the bounding rectangles of the two objects being analyzed, sorted by increasing size; \( (x_a, y_a) \) and \( (x_b, y_b) \) are the top left corner location of \( a \) and \( b \) and \( \text{area}_s \) is the overlapping area.

Considering that we are verifying if the rectangles should be rejected, letting \( \text{area}_a \) be the area of rectangle \( a \), which is the rectangle with smaller area, and \( \beta \) an arbitrary value chosen by the user in the interval [0, 1], we have:

\[
\text{area}_a > \text{area}_s, \beta \quad (9)
\]

The overlapped area, \( \text{area}_s \), is thus compared with \( \text{area}_a \). When (9) is satisfied, the common area is superior to a percentage of the area of this rectangle \( a \), and thus object \( a \) is
rejected, otherwise there is no rejection. Rectangle $b$ is never rejected as it is assumed that the bigger rectangle contains the main body of the vehicle and the smaller rectangle may be other vehicle near vehicle $b$ or a detail of the vehicle $b$, which in this case must be ignored in the counting.

**Oclusion detection** – The detection of occlusion vehicles was made using multiple VDLs. The proposed approach is partially based on the method presented in [19]. The method relies on drawing multiple VDLs in different positions along the same lane and tries to find a correspondence between the vehicles identified in the different resulting TSIs, checking for vehicles that may not be occluded in some of the TSIs to detect them.

The main difference between the proposed method used and the one presented by Mithun et al. [19] is the way the vehicles are associated between different TSIs.

The method proposed here is based on the position of the vehicles observed in one TSI’s VDL and the expected position in the other TSI, which is estimated using the known positions of the VDL relative to each other. With this knowledge, it is possible to correctly associate the vehicles in both TSIs and signal possible undesired objects found throughout the TSIs, for instance due to noise. This avoids early erroneous occlusion detection of vehicles, assuring that the remaining objects, which do not have a correspondence to one of the TSIs, are either occluded vehicles or other moving objects detect in the TSI. The final result of the vehicle association is illustrated on Fig. 7, where the white rectangle represents a detected occlusion.

![Fig. 7: Occlusion detection using 2 TSIs](image)

**Vehicle Counting** – The final step of the proposed system consists of computing the final number of vehicles which were crossing the selected lane for the duration of the video sequence, based on the information provided by the TSIs analyzed thus far and their occlusion analysis. Therefore, the final number of vehicles is simply the sum of the vehicle that where associated between the TSIs and the occlusions which were found.

**Cumulative data treatment** – The data acquired from feature extraction from the objects detected, namely their area, height and width, in multiple video sequences can be stored and used to improve the object validation, if the videos are provided by the same static video camera. Hence, this cumulative data of the features acquired throughout distinct video sequences, when available, is used on the average values utilized across the whole system.

On each new video sequence analyzed, a selected set of values of each feature is added to the cumulative data of that featured and is followed by a computation of the cumulative moving average of each feature thus far. In most video sequences, only a portion of values of each feature is added to the cumulative moving average. The accepted values must be in a specified interval to be accepted. This interval is defined by the parameter $\alpha$. The parameter is used to reject a percentage of the feature value found in the current TSI. Hence, from all the feature values collected on the current TSI, the lower and higher $\alpha\%$ of these values is rejected and thus not included in the cumulative moving average.

The reason for this rejection is that in case of small values, they should be avoided because they may be extracted from objects which are just noise, or in case of big values, the objects may contain occluded vehicles.

The cumulative moving data (CMA) of each feature is afterwards computed using the following expression:

$$CMA_n = \frac{x_1 + ... + x_n}{n}$$  \hspace{1cm} (10)

Where $n$ is the number of values added to the CMA and $x$ is the value of a certain feature in a certain object.

This cumulative data is used instead of only the data provided by the current video sequence when $n$ is above a specific value. By default, the cumulative data is used when $n$ is over 25. The assumption made is that over such a value of $n$, the features averages provided by the cumulative data can provide better results, when used on FVE, than only the feature averages provided by the feature values found on the current video being analyzed.

**IV. EXPERIMENTAL RESULTS**

The parameter used here to evaluate the performance of the implemented system is its vehicle counting accuracy. In addition, an efficiency analysis of the occlusion detection approach used on the developed system is included. The principal causes that reduce the accuracy of the vehicles counts are:

- noise present on the video sequence due to the poor image quality provided by the LTSV cameras, causing false positive detections;
- operational conditions such as rain, bad lighting or darkness which impose lower contrasts between the roads and the moving vehicles and thus may lower the vehicle detected;
- LTSV imperfections due to video acquisition or transmission problems which cause jitter or blur to the frames of the video.

**A. Day Mode Results**

Table 1 summarizes the obtained vehicle count accuracies obtained by the developed system when operating in the day mode. The video sequences used consider variable traffic conditions, from low traffic/empty roads to high/congested traffic. This table provides information relative to the number of video sequences used and the number of vehicles present on each of the analyzed VDLs.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Accuracy</th>
<th>Ground truth</th>
<th>Video sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good illumination</td>
<td>98.7%</td>
<td>155</td>
<td>20</td>
</tr>
<tr>
<td>Jitter</td>
<td>94.6%</td>
<td>146</td>
<td>20</td>
</tr>
<tr>
<td>Shadowed area</td>
<td>95.6%</td>
<td>91</td>
<td>10</td>
</tr>
<tr>
<td>Raining</td>
<td>91.8%</td>
<td>98</td>
<td>11</td>
</tr>
<tr>
<td>Blurred</td>
<td>94.2%</td>
<td>206</td>
<td>20</td>
</tr>
<tr>
<td>Shaking camera</td>
<td>90.0%</td>
<td>129</td>
<td>17</td>
</tr>
<tr>
<td>Dusk</td>
<td>92.2%</td>
<td>207</td>
<td>20</td>
</tr>
<tr>
<td>Averages/Totals:</td>
<td>93.8%</td>
<td>1032</td>
<td>118</td>
</tr>
</tbody>
</table>

Table 1: Day mode results
As expected, the accuracy is higher in video sequences with good illumination as the contrast between the vehicles and the background tends to be higher creating precise object detection. The video accuracy decreases in videos with jitter mainly due to double detections of the same vehicle caused by the jitter effect on the frames and subsequently on the. The videos with shadowed areas present, as main accuracy limitation, the selection of the best Canny threshold values, which allow the detection of vehicles simultaneously on the shadowed and illuminated areas. This value has to be a tradeoff between the best value for detection in each of the areas. The desired performance cannot be always achieved.

The factor which lowered the accuracy in rainy videos was the presence of water drops on the lenses of the cameras, which may move during the video sequences causing dynamic noise and distortions on the generated TSIs. The blurred videos usually occur due to haze on the camera which is a similar situation to the rainy videos. However, this visual obstruction on the lenses does not move significantly during the video sequence and thus has a lower impact on the accuracy. Therefore, the main issue in both rainy video and blurred videos is the loss of contrast due to the lens obstruction, which may result in more undetected vehicle and undesired vehicle occlusions.

The videos captured with a moving camera present the worst accuracy, mainly due to the difficulty to estimate the background of the TSI. When the camera shakes significantly, if the VDL overlaps road markings, these markings tend to be detected as one or more vehicles as the background subtraction cannot completely remove it as background. Therefore, it is preferable to draw the VDLs in areas with no road markings when cameras have a tendency to shake.

The videos captured during dusk have a lower light intensity on the image and the vehicles tend to present shadows due to a lower position of the sun relative to the vehicles. The issues affecting the accuracy are:

- occlusions of vehicles due to a connection of the moving objects by their shadows;
- double detections of the same vehicle due to one object being its shadow and the other its body;
- vehicles which have their headlights already turned on, detecting the reflection of the light on the pavement as another object.

The obtained accuracy is thus mainly affected by the capacity to detect occlusions in the TSIs and the ability to reject the double detections.

The occlusions found when operating in day mode are caused mainly by shadows of vehicle moving close to each other and by blur on the video which reduces the capacity of the system to individualize the vehicles as single object. The occlusion detection by positioning of two parallel VDLs at 5 pixels of each other was able to find and correct 81.1% of the occlusions in the generated TSIs. This proves that using two VDLs for occlusion detection can be used as a tool to find the majority of the occluded vehicles.

### Night Modes Results

Table 2 summarizes the obtained vehicle counting results for the proposed system when operating in one of the considered night modes: Night Back Side Mode (NFSM) or Night Back Side Mode (NBSM).

<table>
<thead>
<tr>
<th>Video sequences</th>
<th>Averages/Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBSM – Good illumination</td>
<td>93.6%</td>
</tr>
<tr>
<td>NBSM – Blurred</td>
<td>90.6%</td>
</tr>
<tr>
<td>NBSM – Jitter</td>
<td>90.2%</td>
</tr>
<tr>
<td>NFSM – Good lighting</td>
<td>89.4%</td>
</tr>
<tr>
<td>NFSM – Blurred</td>
<td>90.6%</td>
</tr>
<tr>
<td>NFSM – Jitter</td>
<td>89.0%</td>
</tr>
<tr>
<td>Averages/Totals</td>
<td>90.6%</td>
</tr>
</tbody>
</table>

The obtained accuracies on these two modes are slightly inferior to those obtained when operating in day mode. This is expected due to the lower luminaire levels and resulting lower contrast found in the generated TSIs.

It is interesting to note the low variance in the observed accuracies in most cases, especially in the NFSM mode, where the Blurred videos have actually a slightly better accuracy than the videos captured with good lighting. This shows that the use of an Otsu binarization in the NFSM, in order to just detect the headlights reflection on the pavement, made this operation mode more robust to noise. However, this is also the operation mode with lower accuracies, mainly due to a higher number of false positive detections of vehicles. The false positives detections are typically caused by multiple detections of the same vehicles.

The same approach with VDLs distancing 5 pixels was used on the night operation modes to detect occlusions. On the night modes, the percentage of detected occluded vehicles decreased slightly, to 71.9%, due to the lower capacity to differentiate moving objects at night caused by the lower luminance. The number of occlusions on the NFSM is almost zero due to the different approach using the Otsu’s binarization to just find the reflection of the headlights of the vehicles on the road pavement.

### Cumulative data results

This section reports the performance evaluation computed when considering a set of 16 video sequences, all captured by the same video camera, pointing at the same road lane without any position or angle variation.

Two sets of performance tests were conducted to evaluate the effect of cumulative data treatment. In both cases two VDLs were placed in the exact same coordinates.

For the first set of experiments, 12 videos with good illumination were used. This set of experiments used 11 of the 12 videos for training and used the remaining video to evaluate the results. The process was repeated 12 times in order to test all the possible combinations.

For the second set of experiments, 16 videos were used. The 12 videos with good illumination from the previous set and 4 dusk videos. This set of experiments used 15 of the 16 videos for training and used the remaining video to evaluate the results. The process was repeated 16 times in order to test all the possible combinations. The goal is to verify if using videos
with different operational conditions in the cumulative data can improve the overall accuracy of the system.

All the system parameters were set automatically in both methods. The results of both experiments are presented on Table 3.

<table>
<thead>
<tr>
<th>Sets</th>
<th>Accuracy with untrained data</th>
<th>Accuracy with trained data</th>
<th>Total number of vehicles counted on test videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93.5%</td>
<td>96.9%</td>
<td>91</td>
</tr>
<tr>
<td>2</td>
<td>94.7%</td>
<td>97.2%</td>
<td>127</td>
</tr>
</tbody>
</table>

Table 3: Cumulative data results

The results of the first set of experiments show that the accuracy with trained data tends to slightly improve the vehicle counting accuracy. The results of the second set of experiments also show accuracy improvements compared to the accuracy with untrained data. As a result, we can conclude that the use of cumulative data treatment tend to improve the quality of the results that can later be achieved when analyzing videos, even if captured in different conditions.

D. Computational Performance

The principal element which affects the time and speed of the various steps of the developed system is the spatial and temporal resolutions, and also the duration of the video sequence. This is due to the increase of computational resources necessary as a result of bigger TSIs which need to be analyzed.

The tests reported on this paper were run on a personal desktop computer equipped with an Intel Core i7 CPU, 4 GB of RAM on a 64-bit Ubuntu 12.04 operating system, which can be considered moderate computational resources. The running times presented in Table 4 were obtained considering 2 VDLs, with the second VDL being generated automatically based on the position of the first one. The tests were made on low resolution video sequences of 15 seconds duration, with 15 FPS for the low resolution video sequences and 19 seconds with 30 FPS for the high resolution ones.

<table>
<thead>
<tr>
<th>Video Resolution</th>
<th>Average running time</th>
<th>Average time per frame</th>
<th>Number of runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>200x200</td>
<td>0.357 s</td>
<td>1.47 ms</td>
<td>20</td>
</tr>
<tr>
<td>1920x1080</td>
<td>10.39 s</td>
<td>21.64 ms</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4: Time results

As expected, the running time increases with the resolution of the video. The main reason is the bigger size of the images on the high resolution video sequence. The higher resolution images need more resources from the system to process and to generate the TSIs. The TSIs also tend to have larger dimensions, because there are more pixels detailing each road lane crossed by the VDLs. The high resolution video used has a higher FPS rate than low resolution videos, which also increases the running time. However, the running times are still significantly lower than the duration of the videos being analyzed. On the low resolution videos, they are around 42 times lower and on the high resolution videos, they are around 2 times lower. This still allows the system to be used on real-time situations in both cases.

E. Results Comparison

This section provides a comparison of the performance results of the developed system with other developed works. The literature related with vehicle counting tends to focus on their performance tests on videos with good lighting [9]. Hence, the good lighting accuracy of the developed system is used as comparison of the results on the Table 5 in this section.

The comparisons were made with five other systems. In Rashid et al. [14] a background-subtraction-based (BSB) method with a heuristically chosen threshold is used to individualize and count the vehicles. The systems developed by Hue et al. [20] and Rashid et al. [21] use a single VDL (SVDL) approach for detection of the vehicles, while Mithun et al. [19] and the system developed in this paper use a multiple VDL (MVDL) approach.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Vehicles counted</th>
<th>Ground truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSB [5]</td>
<td>86.2%</td>
<td>306</td>
<td>355</td>
</tr>
<tr>
<td>SVDL [20]</td>
<td>92.7%</td>
<td>329</td>
<td>355</td>
</tr>
<tr>
<td>SVDL [21]</td>
<td>92.7%</td>
<td>329</td>
<td>355</td>
</tr>
<tr>
<td>MVDL [19]</td>
<td>98.3%</td>
<td>349</td>
<td>355</td>
</tr>
<tr>
<td>Proposed system</td>
<td>98.6%</td>
<td>292</td>
<td>296</td>
</tr>
</tbody>
</table>

Table 5: Results comparison

The obtained accuracy of the developed system is clearly superior to the method using a BSB approach and has a significant accuracy improvement compared to the SVDL approach. The system has slightly superior accuracy compared to the MVDL system developed by Mithun et al. [19]. It would be interesting to compare the performance of both systems on night videos, however there were no tests performed on such operational conditions by the authors of the paper.

V. CONCLUSIONS

The developed Video-based Traffic Management system achieves its goals to be efficient and flexible method to count vehicles. The system can be used as an efficient and low cost replacement for other vehicle counting methods.

A. Strengths

The use of more than one VDL to analyze the features extracted from the TSIs, allows the development of a more flexible, accurate and robust Video-based Traffic Management Systems. The main strengths are summarized in the following points:

• there are no limitations to the number of VDLs used on each video nor angle and length limits to them;
• there are no limits for the resolutions of the videos, their duration or their resolution;
• the system has a good performance and robustness in all the operational conditions, especially on day mode video sequences, where it can achieve an accuracy of almost 99% and even on dark videos the lowest accuracies round the 90%;
• The vehicle count is acquired with a limited used of computational resource for videos of any size and duration. The processing time rarely surpasses 0.5 seconds for low
resolution video sequences of 15 seconds, which allow this system to be also used in real time situations.

B. Weaknesses

The performance and efficiency of the developed Video-based Traffic Management System is dependent on the quality of the TSIs it can generate. Hence, the factors which affect the quality of the TSIs are the main weaknesses found in this vehicle counting approach, which are summarized in the following points:

- Cameras with continuous shaking movements cause distortions on the TSI which reduce the efficiency of the background estimation and subtraction. This situation might even turn the vehicle count impossible if the shaking is extremely accentuated;
- The light reflections at night represent a problem in the vehicle detection due to possible double detections by the head lights of the vehicles;

C. Future Work

There are some aspects and features which could be used to improve on the system with further investigation on the use TSIs for vehicle detection and counting. This topic is discussed in the following paragraphs.

The TSIs approach presented here for detection does not take any advantage of the color channels which may be available on the video. In this paper, the TSIs are always converted to a grayscale format for analysis. This color could probably be used, for instance, to detect the red taillights of the vehicles at night and use it as an alternative way to count vehicle in scenario with high tendency for the occurrence of occlusions. This could be especially helpful on occlusions caused by shadows, because the presence of shadows tends to not interfere significantly with the lights emitted by the vehicles.

The width of the objects on the TSI varies accordingly to the speed the object crosses the VDL and also accordingly to the FPS rate of the camera. This width could be used to have an estimate of the velocity of the vehicle. To work correctly, the software needs to have enough long term information of the objects’ features with the camera on a constant angle and position. This could be used to know if the traffic velocity is changing and could help predict possible traffic congestions before it occurs.

REFERENCES