www.ietdl.org

Published in IET Intelligent Transport Systems
Received on 17th March 2011
Revised on 8th August 2012
Accepted on 18th September 2012
doi: 10.1049/iet-its.2012.0020

ISSN 1751-956X

Dual-rate background subtraction approach for estimating traffic queue parameters in urban scenes

Jose M. Milla, Sergio L. Toral, Manuel Vargas, Federico J. Barrero

E. T. S. de Ingeniería, University of Seville, Avda. Camino de los Descubrimientos s/n, 41092, Seville, Spain
E-mail: toral@esi.us.es

Abstract: This study proposes traffic queue-parameter estimation based on background subtraction, by means of an appropriate combination of two background models: a short-term model, very sensitive to moving vehicles, and a long-term model capable of retaining as foreground temporarily stopped vehicles at intersections or traffic lights. Experimental results in typical urban scenes demonstrate the suitability of the proposed approach. Its main advantage is the low computational cost, avoiding specific motion detection algorithms or post-processing operations after foreground vehicle detection.

1 Introduction

The growth in traffic congestion has been recognised as a serious problem in many urban areas. Road transport has become by far the major source of environmental degradation in urban centres, where most of the population live and work. The massive contribution of traffic to the global environmental problem is likely to become even more critical in the future, in view of the threat of continuous growth in the demand for transport [1, 2]. In this context, transport and planning policies may have a considerable effect on traffic emissions. However, these policies must be guided by the measurement of traffic parameters in general and queue parameters in particular [3].

Among the large number of research areas related to intelligent transportation systems, the vision-based approach has become one of the main streams of research [4]. A vision-based traffic monitoring system performs automatic traffic parameter estimation with the help of image processing techniques, sending this information to the traffic control centre to provide real-time traffic information to drivers and to take decisions about traffic management. The main advantages of computer vision methods over traditional techniques are flexibility, ease of installation and maintenance. However, they are very sensitive to changing ambient light conditions and shadows.

In recent years, a huge research effort has been focused on the application of image processing techniques to automatic traffic-parameter estimation. The early works in this field proposed a window-based edge detection technique to measure traffic volume, types of vehicles, queue parameters and movements of vehicles [5, 6]. A feature-based tracking system for detecting and tracking vehicles was proposed in [7]. This algorithm can measure many micro-scaled parameters such as instantaneous state of velocity. However, the algorithm exhibits some problems when vehicles are moving in multiple directions through the detection region. An entropy-based approach to detect vehicles and extract traffic parameters was presented in [8]. They used inter-frame differencing to extract the active pixels of the moving vehicles, and the exponential entropy as the basic measurement for vehicle detection, whenever its value exceeds a given threshold. A mixture of texture-based vehicle segmentation exponential entropy-based vehicle detection was developed in [9] for a traffic surveillance system at an urban intersection. Background subtraction is also a very common technique for detecting moving objects from image sequences using a static camera. The idea consists of extracting moving objects as the foreground elements obtained from the ‘difference’ image between each frame and the so-called background model of the scene [10, 11]. Background subtraction techniques have also been used for traffic parameter extraction in [12]. Some refinements of this technique for achieving computationally efficient implementations and urban context adapted implementations can be found in [13, 14], respectively.

Although the majority of works related to traffic parameter estimation are mainly focused on detection and tracking of vehicles, a vision-based traffic monitoring system can also be very useful in determining the length of a traffic queue (typically, in front of a traffic light or intersection), without relying on segmenting individual vehicles. In this paper, a queue estimation algorithm based on the combination of two background models of the scene is proposed. The first model is programmed with the aim of being very sensitive to changes in the scene, so temporarily stopped vehicles are incorporated to this background, whereas the second one is programmed with a lower adaptation speed, so temporarily stopped vehicles remain on the foreground. The proposed algorithm avoids the use of specific motion detection algorithms included in previously reported works. The rest of the paper is organised as follows: next section provides an overview of previous works related to traffic queue estimation. Section 3 presents the proposed queue estimator algorithm, detailing the short-term and long-term background models and the set of queue parameters being estimated. Experimental results are described in Section 4, considering four sequences coming from both, personal and publicly available datasets, and including a night-time sequence. Finally, conclusions are presented in Section 5.

2 Related work

The main challenge of traffic queue estimation is the fact that vehicles are temporarily stopped. Thus, vehicle segmentation algorithms have to rely on previous trajectories of individual vehicles or on three-dimensional (3D) models for this purpose. To the best of our knowledge, just few works are specifically devoted to traffic queue parameter estimation. A combination of vehicle presence and motion detection algorithms is presented in [6]. A motion-detection operation based on an edge detection algorithm is first applied, and then, if the algorithm detects no motion, the vehicle-detection operation is used to decide whether there is a queue or not. Next, they used a trained neural network as a classifier to judge the length of a traffic queue. However, the adopted neural network has to be trained extensively for various scenes and situations. A macroscopic method to detect the length of a traffic queue was proposed in [15]. An average operation is applied to ten consecutive frames to remove moving vehicles, and then the road section is divided into several unit areas to decide whether or not there are vehicles on each, depending on a predetermined grey value threshold. However, this method exhibits two major problems. First, averaging frames to detect and remove moving vehicles would not provide good results in scenarios with vehicles moving with different speeds. Second, the decision criterion for vehicles temporarily stopped based on a grey value threshold is too simple when considering the variety of scenarios and illumination conditions in urban scenes. Background subtraction has been used in [16] to detect traffic light cycle failures, which happens when one or more queued vehicles are unable to depart because of insufficient capacity during a signal cycle. In that work, an initial supervised background estimation is required during a given training period of around 60–90 frames (working at a lowered frame rate of 4 fps). A simple median filter is used for background subtraction, which requires that each background pixel is clearly visible for at least 50% of the time. The supervision is required to make sure, by human means, that this assumption holds during the initial training period. This same restriction is the reason why this basic algorithm is unsuitable for urban traffic scenes, in the long term. What the authors in [16] propose in the long term is to dynamically update this initial background model by an elementary forgetting-factor rule. Another critical point of this algorithm is the estimation of the end-of-queue with 'motion images', which are obtained as the plain absolute difference between two consecutive frames. The motion image corresponding to the current frame is then combined with the result of background subtraction with respect to the dynamically updated background model, in order to detect the stopped vehicles in the queue. Even assuming that these simple inter-frame differences succeed extracting objects moving at different speeds, the whole concept is based on the idea that the background image is able to avoid merging the stopped vehicles in a queue. However, this requirement is in conflict with the need of a background model being promptly adaptive to relatively sudden changes in the illumination conditions of the scene, as the background has to be updated very slowly.

In this paper, the idea of combining background models that can be updated with different speeds is exploited to achieve the detection of temporarily stopped vehicles. A similar approach has been proposed in [17] but with a different application: abandoned item and illegally parked vehicle detection. In this case, both background models are obtained by a mixture of Gaussians, which has been reported to have some limitations in dynamic scenes with strong variations or non-stationary properties [18]. The main contributions of the paper are (i) the proposal of a queue detection estimator that makes use of typical methods for vehicle detection (background subtraction), improving the detection method proposed in [15]. (ii) The proposed approach allows the segmentation of temporarily stopped vehicles, without relying on any subsequent vehicle tracking algorithm, like in [6]. (iii) The proposed approach, based on a sigma–delta filter implementation of the background subtraction algorithm, is very computationally efficient, and can be easily implemented on embedded processors.

3 Proposed queue detection algorithm

A vehicle queue is characterised by the presence of vehicles with no motion. The problem with the classical background subtraction is that temporarily stopped vehicles can be considered either as foreground or background depending on the time the vehicle has been stopped and the updating frequency of the background model [19]. The approach proposed in this paper considers two background models with different updating speeds.

A short-term background model must adapt quickly to changes in the scene. Consequently, those vehicle temporarily stopped are right away integrated in the background model. A 'sigma–delta background subtraction algorithm has been chosen because of its high computational efficiency [13].

A long-term background model must adapt to changes slowly, preserving the background model from those vehicles which are temporarily stopped. The idea is preserving the model from being corrupted with slow-moving vehicles or vehicles that are motionless for a time gap [20, 12].

Both models are combined according to the operations detailed in the block diagram of Fig. 1. Starting from the current frame, the basic sigma–delta algorithm provides a background model for moving vehicles (short-term background), whereas an enhanced version provides a background for vehicle presence (long-term background), also signalling as foreground any temporarily stopped vehicle.

For each new frame, the background subtraction isolates pixels belonging to the target objects for each algorithm. A NOT operation is applied to the basic sigma–delta subtraction, providing all the background objects, including those recently stopped. The common pixels from this image and the enhanced sigma–delta background subtraction (AND operation in the block diagram) provide the so called 'still-presence image', which signals any temporarily stopped object as foreground.

3.1 Short-term background model

The sigma–delta background estimation algorithm provides a recursive computation of a valid background model of the

doi: 10.1049/iet-its.2012.0020

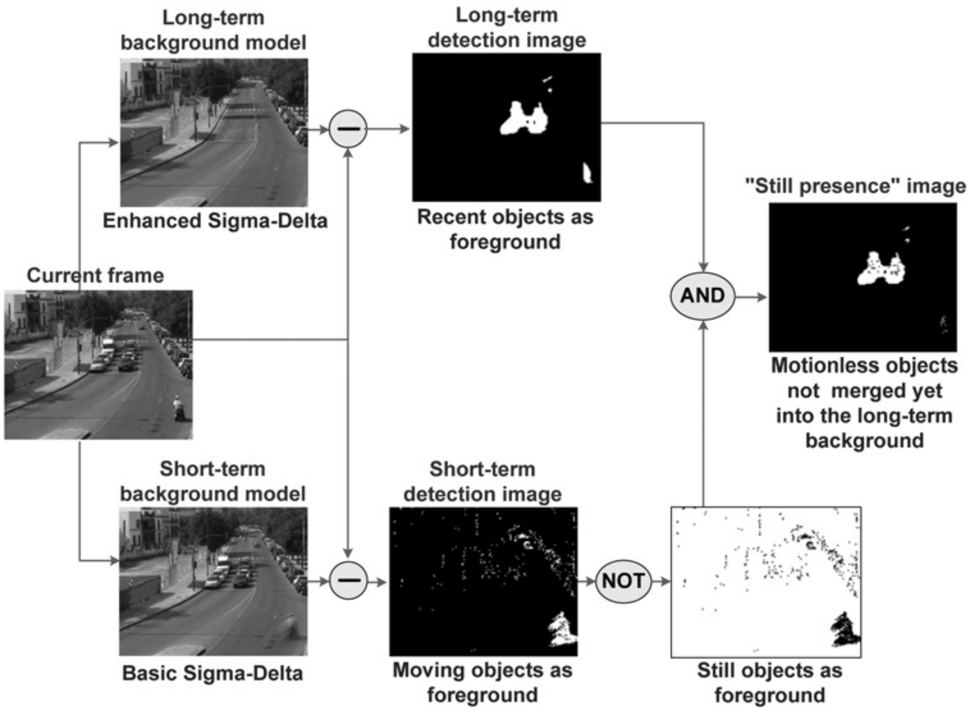

Fig. 1 *Block diagram of the proposed queue detection algorithm*

scene assuming that, at the pixel level, the background intensities are present most of the time [18]. Fig. 2 describes the basic sigma–delta algorithm. For readability purposes, the syntax has been compacted in the sense that any operation involving an image should be interpreted as an operation for each individual pixel in that image.

I_t represents the current input image, M_t represents the background-model image at frame t and V_t represents the temporal variance estimator image (or variance image, for short), holding information about the variability of the intensity values at each pixel. It is used as an adaptive threshold to be compared with the difference image. Pixels with higher intensity fluctuations will be less sensitive, whereas pixels with steadier intensities will signal detection upon lower differences.

The only parameter to be adjusted is N, with typical values between 1 and 4. In particular, $N = 4$ has been chosen in the provided experiments. Another implicit parameter in the algorithm is the updating period of the statistics, which depends on the frame rate and the number of grey levels. This updating period can be modified by performing the loop processing every P frames, instead of every frame. The same algorithm computes the detection image or detection mask, D_t. This binary image highlights pixels belonging to the detected foreground objects (one-valued pixels) in contrast to the stationary background pixels (zero-valued pixels). The described algorithm is, in fact, a slight variation of the basic sigma–delta algorithm, where the background model is only updated for those pixels where no detection is signalled, instead of doing it for all pixels. This selective updating is called 'relevance feedback' and it is usually preferable, as it provides more stability to the background model.

$M_o = I_0$ // *Initialise background model M*
$V_o = 0$ // *Initialise variance V*
for each frame t
  $\Delta_t = |M_t - I_t|$ // *Compute current difference*
  if $\Delta_t \neq 0$
    $V_t = V_{t-1} + \mathrm{sgn}(N \cdot \Delta_t - V_{t-1})$ // *Update variance V*
  end if
  $D_t = (\Delta_t \geq V_t)$ // *Compute detection image D*
  if $D_t = 0$ // *Update background model M ...*
    $M_t = M_{t-1} + \mathrm{sgn}(I_t - M_{t-1})$ // *with relevance feedback*
  end if
end for

Fig. 2 *Basic sigma–delta background estimation*

The main advantage of the sigma–delta algorithm is its ability of quickly providing valid background models whenever this background is present on the scene most of the time. However, this advantage is also its main disadvantage when objects are temporarily stopped. For instance, vehicles in front of a traffic light are also promptly incorporated into the background model.

Fig. 3 illustrates this situation. As vehicles are obtaining stopped in front of the traffic light, they are integrated into the background model. Although this is an undesirable effect for vehicle-detection applications, it will be advantageous for our purposes. The background subtraction will only retain as foreground those moving vehicles.

3.2 Long-term background model

The purpose of the long-term background model is to avoid the problems of the previous algorithm regarding vehicles which are motionless for a time gap. Basically, it is an enhanced version of the basic sigma–delta algorithm, introducing a numerical confidence level attached to each pixel in the current background model. The algorithm is detailed in Fig. 4, using the basic sigma–delta algorithm as a starting point [20].

As we can see, in addition to the same parameter N described in the previous case, there is now a new set of

doi: 10.1049/iet-its.2012.0020

Fig. 3 *Incorporation of stopped vehicles in front of a traffic light (I: current frame, M: background model)*

parameters determining thresholds and initial conditions. The following values have been chosen for them: $v_{\text{ini}} = v_{\text{min}} = 10$, $c_{\text{ini}} = c_{\text{min}} = 10$.

In the proposed algorithm, the variance image is intended to represent the variability of pixel intensities when no objects are over that pixel. In other words, the variance image will solely model the background intensities, as a proper threshold should be chosen from that. A low variance will be interpreted as having a 'stable-background model', that should be maintained. A high variance will be interpreted as 'the algorithm should look for a stable-background model'. One of the problems of the previous versions of sigma–delta algorithms in urban traffic environments is that, as the variance grows when vehicles are passing by, the detection degrades because the threshold becomes too high. Then, it is necessary to perform a selective background and variance update. Both should be updated only when the traffic conditions are presumed to be suitable. For this purpose, the algorithm evaluates the behaviour of each pixel's intensity during a given period, called 'confidence period'. At the end of each period, the algorithm calculates the number of detections during that period; the ratio of detections is taken as an estimation of

$M_o = I_0$, $V_o = \nu_{ini}$ *// Initialise background model, variance*
$I_0^{DC} = I_0^{FC} = 0$, $I_0^{CON} = c_{ini}$ *// Initialise detection and frame counter*
for each frame t *// and confidence measure*
    $I_t^{FC} = I_t^{FC} + 1$ *// Increment frame-counter image*
    if $I_t^{FC} \geq I_t^{CON}$ *// If confidence period expires*
        <evaluateConfidencePeriod>$(I_t^{FC}, I_t^{DC}, I_t^{CON}, U_t)$
        if $I_t^{CON} == c_{\min}$ *// If confidence reaches the minimum ...*
            $U_t = 1$ *// force updating*
        endif
        $I_t^{DC} = I_t^{FC} = 0$ *// Reset detection counter and frame counter*
    else if I_t^{FC} **is a multiple of** P *// If refresh period expires*
        if $\left(I_t^{DC} / I_t^{FC}\right) \leq 0.2$ *// Very light traffic*
            $U_t = 1$ *// Force updating disregarding variance*
        end if
    endif
        if $U_t == 1$ *// If updating recommended, sigma-delta algorithm*
        $M_t = M_{t-1} + \text{sgn}(I_t - M_{t-1})$ *// Update background model M*

Fig. 4 *Proposed algorithm: sigma–delta with confidence measurement*

the traffic intensity level over that pixel. Notice that this is an acceptable assumption if we assume that the threshold filters out background intensity fluctuations, as intended. Depending on this ratio, different actions are taken over the confidence measurement, and a decision about the convenience of updating the background model at that time is made. These heuristics are detailed in Fig. 5.

As we can see, for the evaluation of the confidence period, the key parameters are those used to partition the feasible range for the detection ratio $\left(I_t^{\text{DC}}/I_t^{\text{FC}} \in [0..1]\right)$. This partitioning is a way of discriminating between different qualitative traffic conditions with rather fuzzy boundaries: 'very light', 'light', 'moderate', 'heavy' and 'very heavy'. After some experimentation, it seemed that an unbiased partition of the [0..1] interval was fairly appropriate. Some further experimentation showed that common actions could be taken in some cases, as in the case of 'heavy traffic' and 'very heavy traffic' conditions under low variance circumstances, both leading to the same conservative action with a slight penalty in the confidence measure, as the priority is saving the current background. Nevertheless, for the sake of symmetry, the five-interval subdivision has been maintained in Fig. 5.

In particular, in a situation where vehicles are stopped because of a traffic congestion or a red light (high detection rate), and the current-background model has a high confidence measurement, the algorithm would decide not to update the background model and to slightly decrease the confidence measurement. The latter action is convenient to avoid the model to obtain indefinitely locked in a wrong or obsolete background.

The confidence measurement is related to the maximum updating period. In very adverse traffic conditions, this period is related to the time the background model is able to keep untainted from the foreground objects. The detailed description of the enhanced sigma–delta algorithm can be found in [20].

With respect to that algorithm, a new mechanism has been included, with the aim of achieving some degree of adaptability to global illumination changes, because of a true variation in the ambient conditions or to the auto-iris mechanism of the own camera. As a consequence of this mechanism, the current background image is adapted to the global illumination level of each new acquired frame, before performing background subtraction via the sigma–delta algorithm. This adaptation of the background model is implemented through a linear intensity mapping. This mapping is recomputed with each new frame in the following way: for every pixel signalled as background in

<evaluateConfidencePeriod>$(I_t^{FC}, I_t^{DC}, I_t^{CON}, U_t)$
if $V_t \le v_{\min}$ *// Low variance => priority: keep current background*
if $\left(I_t^{DC}/I_t^{FC}\right) \le 0.2$ *// Very light traffic*
$I_t^{CON} += 10$ *// Significantly increase confidence measure*
$U_t = 1$ *// Background model can be updated (low risk)*
else if $0.2 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.4$ *// Light traffic*
$I_t^{CON} += 0$ *// Do not increase/decrease confidence measure*
$U_t = 1$ *// Background model can be updated (low risk)*
else if $0.4 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.6$ *// Moderate traffic*
$I_t^{CON} += 0$ *// Do not increase/decrease confidence measure*
$U_t = 0$ *// Background should not be updated (medium risk)*
else if $0.6 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.8$ *// Heavy traffic*
$I_t^{CON} -= 1$ *// Decrease confidence measure*
$U_t = 0$ *// Background should not be updated (high risk)*
else if $0.8 < \left(I_t^{DC}/I_t^{FC}\right) \le 1.0$ *// Very heavy traffic*
$I_t^{CON} -= 1$ *// Decrease confidence measure*
$U_t = 0$ *// Background should not be updated (high risk)*
end if

else *// High variance => priority: look for new background*
if $\left(I_t^{DC}/I_t^{FC}\right) \le 0.2$ *// Very light traffic: take advantage of it*
$I_t^{CON} -= 10$ *// Significantly decrease confidence measure*
$U_t = 1$ *// Background model must be updated*
else if $0.2 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.4$ *// Light traffic*
$I_t^{CON} -= 1$ *// Decrease confidence measure*
$U_t = 1$ *// Background model must be updated*
else if $0.4 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.6$ *// Moderate traffic: update*
$I_t^{CON} -= 1$ *// Decrease confidence measure*
$U_t = 1$ *// Background should be updated*
else if $0.6 < \left(I_t^{DC}/I_t^{FC}\right) \le 0.8$ *// Heavy traffic: wait better conditions*
$I_t^{CON} -= 1$ *// Decrease confidence measure*
$U_t = 0$ *// Background should not be updated*
else if $0.6 < \left(I_t^{DC}/I_t^{FC}\right) \le 1.0$ *// Very heavy traffic: wait*
$I_t^{CON} -= 10$ *// Significantly decrease confidence measure*
$U_t = 0$ *// Background should not be updated*
end if

Fig. 5 *Heuristics for the evaluation of the confidence period*

the previous frame (zero-valued pixels in the detection image, D_t), the intensity value of the pixel in the current-background model and the intensity value of the same pixel in the current frame are used as coordinates (x and y, respectively) for a point to be drawn in a 2D plot. Next, a linear regression is estimated from all those points and used as a transformation rule for a intensity mapping. Finally, this mapping can be configured in a look-at table to be applied to every pixel in the current-background model. Unlike [16], with this strategy, we decouple and make compatible the requirement of a long-term background model with relatively low updating speed and the need of an algorithm being adaptive to sudden illumination changes in the scene.

Regarding to local illumination changes, mainly because of moving cast shadows, the eventual problems can be alleviated in a further step by employing techniques related to shadow removal, which is beyond the scope of this paper [19, 20].

The behaviour of the enhanced version of sigma–delta algorithm is illustrated in Fig. 6, based on the scenario already shown in Fig. 3. In this case, however, the stopped vehicles in front of the traffic light have not been absorbed by the background model (image M).

3.3 Description of the estimated queue parameters

A number of queue parameters are estimated from the previous queue detection algorithm. The queue regions to be considered are manually drawn over the image as polygonal regions, during the scene configuration stage (see Fig. 7). When the algorithm detects a new queue which starts building up from the stopping line, the 'triggered queue' signal is switched on. Nevertheless, a hysteresis mechanism is introduced to avoid false detections caused by sluggish vehicles and preventing excessive fluctuations in the estimated queue length.

Starting from the stopping line, a fraction of the region length is defined as a 'full-queue threshold'. Whenever the vehicle queue reaches this point, a 'full-queue' signal is triggered, which is another instantaneous qualitative state of the queue. On the other side, the 'instantaneous queue length' is defined as a quantitative measure, given in percentage of the full-queue length. According to this, if at any given time, the vehicle queue surpasses the full-queue threshold, the queue length measure will be over 100%.

Fig. 8 illustrates the behaviour of queue detection algorithm. Whenever the 'triggered-queue' state is signalled, the region is displayed in a dark grey colour. Otherwise, the region remains

Fig. 6 *Non-incorporation of stopped vehicles in front of a traffic light (I: current frame, M: background model)*

doi: 10.1049/iet-its.2012.0020

Fig. 7 *Evaluated urban traffic scenarios*

a SE01 sequence
b SE02 sequence
c MIT sequence
d Night-time sequence

Fig. 8 *Queue detection for traffic sequences*

a SE01 sequence
b SE02 sequence
c MIT sequence
d Night-time sequence

in its original light grey. If the 'full-queue' signal is activated, the colour of the complete polygonal region is changed to a different tone of grey, in particular, deep grey for the occupied area.

Apart from the previous instantaneous queue parameters, aggregated data can also be measured. Unlike instantaneous parameters, which are given on a frame-by-frame basis, the aggregated data are computed on a traffic-light cycle basis (also referred as 'queue cycle'). In this category, we consider the 'peak queue length', in percentage of the full-queue threshold, as expected, the 'queue-building rate', given in time units, as it shall be estimated as the time the vehicle queue requires for reaching half of the full-queue threshold, measured from the beginning of the cycle (once the traffic light switches to red). Among the qualitative-type parameters, we can also register the 'triggered-queue' or 'full-queue' conditions on a cycle basis. In this case, the triggered-queue state is switched on for a given cycle (to be referred as 'aggregated triggered-queue' state) if the instantaneous triggered-queue state has been on, at least on one frame during that cycle. The same applies to the full-queue state (referred as 'aggregated full-queue' state).

In addition to the previous parameters, if the full-queue threshold is configured appropriately, according to the expected traffic flow or traffic dynamics and the traffic-light period at the particular spot, the full-queue signal could be used as a probabilistic indicator of 'cycle failure risk' (or even 'congestion risk'), which also falls in the category of aggregated data. On top of that, potentially for every cycle, the 'queue triggering time' or 'full-queue triggering time' can be recorded. Table 1 summarises all the described parameters.

4 Performance analysis

Four different sequences have been selected to illustrate the proposed queue detector algorithm (Fig. 7). The first sequence, SE01, shows two frontal lanes where queues of vehicles are formed in front of the traffic light. The second sequence, SE02, provides a rear view of vehicles on a traffic lane, which is a challenging configuration usually avoided during the systems' setup. The third sequence, comprising three lanes, corresponds to the Massachusetts Institute of Technology (MIT) traffic data set, which is publicly available at http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html for research on activity analysis and crowded scenes [21]. Finally, the fourth sequence comes from a night-time traffic scene, covering four lanes. This amounts for ~5 h and 42 min of video recording, entailing up to 226 red-light cycles.

Table 1 Queue parameter classification

Category	Type	Parameter	States	Units
Instantaneous (frame basis)	qualitative	triggered queue	on/off	
		full queue	on/off	
	quantitative	queue length		%
Aggregated (cycle basis)	qualitative	triggered queue	on/off	
		full queue	on/off	
		cycle failure risk	on/off	
	quantitative	peak queue length		%
		queue building rate		s
		queue triggering time		s
		full-queue triggering time		s

The performance of the proposed queue parameter estimation algorithm is compared with a ground-truth dataset, based on human visual inspection (around 80 h of human effort have been required for the whole ground-truth data gathering).

To evaluate the performance in the estimation of the qualitative-type parameters, precision and recall measures are derived. First, we evaluate the qualitative parameters in the instantaneous category. We start classifying every single frame as one of the following, according to a given parameter, for example, 'full-queue' state:

- *True positives (TP):* the algorithm correctly states full-queue ON for that frame.
- *True negatives (TN):* the algorithm correctly states full-queue OFF for that frame.
- *False positives (FP):* the algorithm incorrectly states full-queue ON for that frame.
- *False negatives (FN):* the algorithm incorrectly states full-queue OFF for that frame.

The following quality measures can be derived:

$$\text{Correctness: } Q_{\text{corr}} = \frac{\text{TP}}{\text{Total of estimated positives}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Completeness: } Q_{\text{compl}} = \frac{\text{TP}}{\text{Total of actual positives}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

These are equivalent to the classic concepts known as 'precision' and 'recall', respectively, but using more intuitive names, as suggested in [22] and [23].

For a perfect classification or detection problem, values of Q_{corr} and Q_{compl} will reach their maximum value, 1.

We could take this same performance evaluation to the cycle level, using the qualitative parameters 'aggregated full-queue state' and 'aggregated triggered-queue' state. In this case, we would redefine a true positive as: 'any cycle for which the algorithm states 'aggregated full-queue' ON, and so does the ground truth, even if the 'instantaneous full-queue' event happened at a different frame with respect to the ground truth or during a different number of frames'. Similarly for TN, FP and FN, for large data collection, this less fine-grained evaluation of a detection algorithm can be perfectly valid.

For the instantaneous and aggregated qualitative parameters, Table 2 gives the performance analysis, including both analysis on frame basis and analysis on cycle basis.

The 'cycle-failure risk' has not been included in the table, since, as stated previously, it is a state that can be readily

IET Intell. Transp. Syst., 2013, Vol. 7, Iss. 1, pp. 122–130
doi: 10.1049/iet-its.2012.0020

Table 2 Performance evaluation of the qualitative queue parameters on a frame basis and on a cycle basis

Category	Parameter	TP	TN	FP	FN	Q_{corr}	Q_{comp}
instantaneous (frame basis)	triggered queue	101 033	440 914	29 608	22 882	0.77	0.82
	full queue	28 722	546 344	8651	10720	0.77	0.73
aggretated (cycle basis)	triggered queue	144	61	18	3	0.89	0.98
	full queue	49	163	13	1	0.79	0.98

Table 3 Performance evaluation of the quantitative queue parameters

Category	Parameter	Error	Error TP	Units
instantaneous (frame basis)	queue length	0.93	0.59	%
aggregated (cycle basis)	peak queue length	14.12	8.99	%
	queue building rate	18.27	5.5	s
	queue triggering time	7.23	5.88	s
	full-queue triggering time	8.8	4.28	s

derived from the ‘aggregated full-queue’ parameter, assuming that the full-queue threshold has been configured properly.

Regarding to the quantitative parameters, the corresponding mean errors with respect to the ground truth measures are used and presented in Table 3. Only relevant cycles are used in the mean computations. That is, any cycle with neither true queue built nor queue detected by the algorithm is considered (although it would reduce the mean error values if reckoned). These correspond to the true negatives in the ‘triggered-queue’ state in Table 2 (instantaneous or aggregated, correspondingly with the category of the quantitative parameter on hand). Another observation is that, for those parameters given in time units, the error has to be understood as an absolute delay or time shift.

In the error column, all the relevant samples have been evaluated (only excluding TN as explained). For instance, in the case of the delay in the ‘queue triggering time’, we can imagine a situation in which, according to the ground truth, a vehicle queue existed for a given cycle, but the algorithm detected no queue at all. According to the ‘aggregated triggered queue’ parameter, this will correspond with a FN. In this case, there is a severe penalisation in the mean error computation, since the delay is taken as the remaining traffic-light cycle time, which explains why some delays are so high. It can be arguable whether or not these FNs or, similarly, the FP, should be accounted for, as they have already been accounted for in the analysis of the qualitative data. In any case, for the sake of completeness, we also include an error column where only the TP are considered.

As mentioned in Section 3.3, there is a hysteresis mechanism, introduced with the aim of avoiding excessive fluctuations in the queue length or a large number of irrelevant queue-triggering switches. The downside of this mechanism is that there will be a certain level of error inherently bound to it: queue-triggering delays, delayed update in queue-length estimations etc. An exhaustive analysis of the influence of this factor upon the proposed performance metrics is beyond the scope and extent of the present paper.

The algorithm has been implemented on a standard ‘Intel Core2 Duo CPU’ at 2, 3 GHz, with 3 GB of RAM, under ‘Windows’ operating system. For the software implementation, C++ programming language has been chosen. Full resolution, grey-scale images of an average resolution 728×540 pixels are reduced to 1/16th of their original resolution before being processed. The average time required for processing each frame is 40 ms, providing around 25 fps of processing rate.

5 Conclusion

This paper proposes a new algorithm for queue-parameter estimation using the combination of two background models with different sensitivity to temporarily stopped vehicles. The described logical operations between them allow the segmentation of an image containing these recently stopped vehicles, precluding those objects either permanently stopped or undergoing any motion. The proposed algorithm has been tested on four different scenarios, during a total number of 226 queue cycles and evaluated against a human-gathered ground truth. The provided results demonstrate the accuracy of the proposed approach, based only on background subtraction techniques, hence, avoiding the need of subsequent object tracking strategies.

6 Acknowledgment

The authors gratefully acknowledge the support provided by Spanish Ministry of Education and Science (National Research Project with reference DPI2010-19154) and the Consejería de Innovación, Ciencia y Empresa (Research Project with reference P07-TIC-02621).

7 References

1 Chiquetto, S., Mackett, R.: ‘Modelling the effects of transport policies on air pollution’, *Trans. Air Pollut.*, 1995, **169**, (1–3), pp. 265–271

2 Barrero, F., Toral, S., Vargas, M., Milla, J.M., Cortés, F.: ‘Internet in the development of future road-traffic control systems’, *Internet Res.*, 2010, **20**, (2), pp. 154–168

3 Zanin, M., Messelodi, S., Modena, C.M.: ‘An efficient vehicle queue detection system based on image processing’. Proc. 12th Int. Conf. on Image Analysis and Processing, 2003, vol. 17–19, pp. 232–237

4 Toral, S.L., Martínez-Torres, M.R., Barrero, F., Arahal, M.R.: ‘Current paradigms in intelligent transportation systems’, *IET Intell. Transp. Syst.*, 2010, **4**, (3), pp. 201–211

5 Fathy, M., Siyal, M.Y.: ‘A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters’, *IEEE Trans. Veh. Technol.*, 1998, **47**, (4), pp. 1342–1349

6 Siyal, M.Y., Fathy, M.: ‘A neural-vision based approach to measure traffic queue parameters in real-time’, *Pattern Recognit. Lett.*, 1993, **20**, (8), pp. 761–770

7 Coifman, B., Beymer, D., Mclauchlan, P., Malik, J.: 'A real-time computer vision system for vehicle tracking and traffic surveillance', *Transp. Res. C, Emerg. Technol.*, 1998, **6**, pp. 271–288
8 Hsu, W.-L., Liao, H.-Y.M., Jeng, B.-S., Fan, K.-C.: 'Real-time traffic parameter extraction using entropy', *IEE Proc., Vis. Image Signal Process.*, 2004, **151**, (3), pp. 194–202
9 Zou, Y., Shi, H., Wang, Y., Shi, G., Zhao, H.: 'A robust traffic parameter extraction approach for surveillance system at urban intersection'. Proc. Second Int. Congress on Image and Signal Processing. CISP '09, 2009, pp. 1–5
10 Spagnolo, P., Orazio, T.D., Leo, M., Distante, A.: 'Moving object segmentation by background subtraction and temporal analysis', *Image Vis. Comput.*, 2006, **24**, (5), pp. 411–423
11 Piccardi, M.: 'Background subtraction techniques: a review'. Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, 2004, vol. 4, pp. 3099–3104
12 Toral, S.L., Vargas, M., Barrero, F.: 'Embedded multimedia processors for road-traffic parameter estimation', *Computer*, 2009, **42**, (12), pp. 61–68
13 Manzanera, A., Richefeu, J.C.: 'A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation'. Proc. ICVGIP, 2004, pp. 46–51
14 Vargas, M., Milla, J.M., Toral, S., Barrero, F.: 'An enhanced background estimation algorithm for vehicle detection in urban traffic scenes', *IEEE Trans. Veh. Technol.*, 2010, **59**, (8), pp. 3694–3709
15 Iwasaki, Y.: 'An image processing system to measure vehicular queues and an adaptive traffic signal control by using the information of the queues'. Proc. IEEE Int. Conf. on Intelligent Transportation Systems, 1998, pp. 310–313
16 Zheng, J., Wang, Y., Nihan, N.L., Hallenbeck, M.E.: 'Detecting cycle failures at signalized intersections using video image processing', *Comput.-Aided Civ. Infrastruct. Eng.*, 2006, **21**, pp. 425–435
17 Porikli, F.: 'Detection of temporarily static regions by processing video at different frame rates'. Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, 2007, pp. 236–241
18 Elgammal, A., Harwood, D., Davis, L.: 'Non-parametric model for background subtraction', *Lect. Notes Comput. Sci.*, 2000, **1843**, pp. 751–767
19 Cheung, S.C., Kamath, C.: 'Robust techniques for background subtraction in urban traffic video'. Proc. Video Communications and Image Processing, SPIE Electronic Imaging, 2004, pp. 1–12
20 Toral, S., Vargas, M., Barrero, F., Ortega, M.G.: 'Improved sigma-delta background estimation for vehicle detection', *Electron. Lett.*, 2009, **45**, (1), pp. 32–34
21 Wang, X., Ma, X., Grimson, E.: 'Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models', *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 2009, **31**, pp. 539–555
22 Wiedemann, C., Heipke, C., Mayer, H., Jamet, O.: 'Empirical evaluation of automatically extracted road axes'. Proc. Workshop on Empirical Evaluation Methods in Computer Vision, 1998, pp. 172–187
23 Lombardi, P., Zanin, M., Messelodi, S.: 'Unified stereovision for ground, road, and obstacle detection'. Proc. IEEE Intelligent Vehicles Symp., 2005, pp. 783–788

doi: 10.1049/iet-its.2012.0020