# Illegal Parking Detection using Gaussian Mixture Model and Kalman Filter

1st Rami Alkhawaji
*Faculty of Computing, Engineering*
and Science
*Staffordshire University*
Stafford, United Kingdom
Rami.Alkhawaji@research.staffs.ac.uk

2nd Mohamed Sedky
*Faculty of Computing, Engineering*
and Science
*Staffordshire University*
Stafford, United Kingdom
M.H.Sedky@staffs.ac.uk

3rd Abdel-Hamid Soliman
*Faculty of Computing, Engineering*
and Science
*Staffordshire University*
Stafford, United Kingdom
A.Suliman@staffs.ac.uk

***Abstract*—Automatic analysis of videos for traffic monitoring has been an area of significant research in the recent past. In this paper, we proposed a system to detect and track illegal vehicle parking using Gaussian Mixture Model and Kalman Filter. i-LIDS dataset is used to test and evaluate the algorithm by comparing the results with the ground truth provided, we have tested the system using 4 full videos from i-LIDS to detect parked vehicle whiten specific area. Region of interest has been used to detect Vehicle parks in a no parking zone over sixty seconds and remains stationary.**

**Within the scope of this work, we highlighted the components of an automated traffic surveillance system, including background modeling, foreground extraction, Kalman filter and Gaussian mixture model.**

***Index Terms*—tracking, detection, kalman, illegal parking,**

## I. Introduction

With the continuous increase in vehicle population, traffic accidents and on-road fatalities have been a major concern in the United Kingdom over the past ten years. According to the road accidents and safety statistics published by Department for transport in the United Kingdom [1], road accidents caused five deaths per day on average in the year of 2013. As a result, there has been a growing demand from both the traffic safety authorities and general public to utilize the state-of-the-art technologies to facilitate greater safety, effective traffic control and monitoring, on-road assistance, etc. In this context, automated vehicle and pedestrian detection, tracking, traffic surveillance, behavioral pattern analysis, and anomaly detection have received significant attention from the research community over the past ten years [2]. This paper will focus on the existing camera and sensing technologies as well as video processing for automated traffic analysis, surveillance and vehicle tracking, emergency event detection, etc.

video sensors provide a reliable, efficient, and scalable approach to different traffic applications due to their low deployment and maintenance cost as well as the wide area of coverage, such as, traffic monitoring and surveillance applications [3], driver assistance [4], on video processing-based traffic flow monitoring and traffic monitoring with the assistance of unmanned aerial vehicles [5], [6].

This paper is structured as follows: Section II components of an automated traffic surveillance system, including foreground extraction and background modeling. Section III demonstrates the proposed system , Section IV experimental results and evaluations, Finally, Section V presents our conclusions.

## II. Components of an Automated Traffic Surveillance System

This section presents an overview of a generic video analysis-based traffic monitoring and surveillance system. Any video camera-based traffic monitoring system that can perform automated incident detection typically comprises multi-stage processing of video data obtained from the traffic sensors. Depending on how the processing is performed, a video-based traffic monitoring system can be divided into two categories: i) top-down approach and ii) bottom-up approach [7]. In a top-down or object-based approach, the first stage involves background modeling and foreground detection, which involves modeling the background (static) scene and isolating any moving object as foreground. This is typically performed based on a statistical modeling approach followed by a few post-processing to preserve connectedness of the moving object. Once the foreground object pixels are detected, they are passed to an object classification module [8], [9], [10]. The objective here is to utilize the previously observed training examples to detect the object class and assign a label. Since this approach involves representing the whole moving object as a collection of pixels, it is called the object-based or top-down approach.

On the other hand, bottom-up approaches involve detection of different parts of an object from the image scene which in turn are utilized to construct a holistic scene representation [11], [12]. A typical bottom-up algorithm will focus on parts of the image to determine whether a partial object is present (e.g. certain body parts of a vehicle or a pedestrian). These types of methods are trained with prior examples of different partial objects which are then analyzed spatially and temporally to obtain an overall scene description. In this section, different components of an automated traffic surveillance system are discussed in detail.

### *A. Foreground Object Extraction*

Moving object (foreground) detection from the background is a key component for a wide-variety of video analytics and

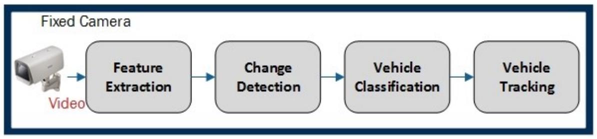

Fig. 1. Current video analytic techniques

motion tracking applications. While the concept of moving object or foreground is trivial to human perception, it is typically treated as a challenging problem in the domain of computer vision. Here, the term foreground refers to any object which is not stationary in a video sequence. However, motion in an image scene can also be caused by camera instead of the object itself, which makes the problem of foreground extraction a non-trivial problem in an uncontrolled environment. Different foreground detection techniques can be found in existing literature, which can be divided into two major categories: i) background modeling based approaches and ii) object appearance-based approaches. In a background modeling approach, the background scene is first modeled when the moving object is not present in the scene. This acts as a generic point of reference when a moving object enters the scene as the foreground pixels could then be easily represented by the error between the new frame and the previously obtained background frame. However, this approach is typically limited to stationary cameras since camera motion can change the background scene, thus rendering the previously constructed background model ineffective. Appearance-based approaches address this issue by utilizing prior knowledge about the moving object to construct an appearance-based representation of that object, which is then tracked over time. This type of approaches typically utilizes different appearance characteristics of the moving object (e.g. color, texture, etc.) to model the object instead of the background. As a result, such approaches can withstand changes occurred from camera movements. From a traffic surveillance and monitoring point of view, foreground object detection based on background modeling is a strong match since prior knowledge about the moving object is often unavailable and the cameras usually remain stationary over time. The following sections discuss both background modeling and object appearance-based foreground object detection.

*1) Frame Differencing:* Frame differencing is one of the most commonly-known foreground extraction method that utilizes the difference between two consecutive frames as a moving object detection mask. Being a simple moving object detection technique, frame differencing offers fast computation, however, suffers under the presence of random noise, non-monotonic illumination variations as well as repeated or periodic appearances of certain objects in the scene. K. Park and D. Lee in [13] utilized frame differencing in their work on the detection of street parking rule violations. Another approach in [14] detected motor-cycles on the road based on frame differencing and multi-modal particle filtering approach. However, since frame-differencing based foreground object detection techniques are susceptible to random noise, image sequences are required to carefully be per-processed to remove noise.

*2) Background Subtraction:* While frame-differencing involves computing the error between two consecutive frames, background subtraction-based approaches focus on constructing a generic representation of the background. Once a background model is constructed, each subsequent frame is compared against the model frame to determine the foreground object. This is typically identified by the error between the two frames which is then thresholded (static or dynamic) to obtain a mask [8]. Different types of background modeling approaches can be found in the literature. The following section provides an overview of a few well-known background modeling approaches.

- **Frame Averaging:** Given a current image sequence frame $f_t$, the frame averaging-based background modeling will consider the mean of all prior frames $f_1, f_2, ..., f_{t-1}$ as the background model. While this approach provides a computationally inexpensive way of modeling the background, it often yields unexpected tracking results due to the contribution of the foreground pixels from the previous frames in the background model. To address this issue, in [8] and [15] proposed to exclude the detected object in the current frame while averaging it with the background model to use for the next frame. This method, named the ”instantaneous background” thus cancels some errors caused by the augmentation of moving object pixels in the background model construction by replacing those pixels with previous background pixels computed for the same location. One of the researcher, in [16] presented a vehicle tracking approach that utilized a similar background modeling approach for cameras placed on lower ground thant typical traffic cameras. However, the selection of frame averaging method was motivated due to the lower computational cost as their proposed method utilizes expensive 3D perspective projection computation for the tracking purpose.
- **Single Gaussian:** The single Gaussian-based background modeling approach extend the frame averaging method by also constructing a separate image representing the variance of the all previous frames along with the mean image. A pixel in the current frame is labeled as foreground or background depending on its position in the Gaussian distribution. From a statistical point of view, the single Gaussian distribution can be compared with a dynamic thresholding technique. B. Morris and M. Trivedi in [17] presented a vehicle classification and tracking approach in their work where they classified a vehicle in the scene as one of the sedan, semi, or suv+truck+pickup class. Here, the single Gaussian method was used for background modeling.
- **Mode of Pixel History:** Another approach of modeling the background is to utilize the mode of the history of a pixel, as proposed by J. Zheng and Y. Wang in [18]. Their non-parametric estimation of the background model

using a fixed-length window on the historical values of a pixel location and then label the most frequently occurred value as part of the background. Intuitively, for some pixel to be considered as the background, the value of that pixel should be the same over a certain period of time. This is a fundamental assumption regrading the property of a background pixel, which is also used in the Gaussian mixture models (GMM). However, unlike the GMM approach, mode estimation-based background modeling is sensitive to the selection of bin size due to its dependence on the presence of a peak or resonant structure in the distribution.

- **Kalman Filtering:** Kalman filtering can also be utilized to construct the background model of an image, where each pixel of the image sequence is modeled based on an individual Kalman filter [19].
- **Wavelets:** [20] presented a wavelet-based background modeling approach where a probability density function is estimated for each pixel of an image based on the Marr wavelet kernel. The proposed method was demonstrated to be robust against gradual lighting changes, while it could detect both moving vehicles and pedestrian in the urban road image scene.

## III. The proposed system

In our case, the system should give an alarm if a vehicle parks in a no parking or stops within a no parking zone and remains stationary for over sixty seconds using Gaussian mixture model and Kalman filter, see Fig. 2 for the no parking zone.

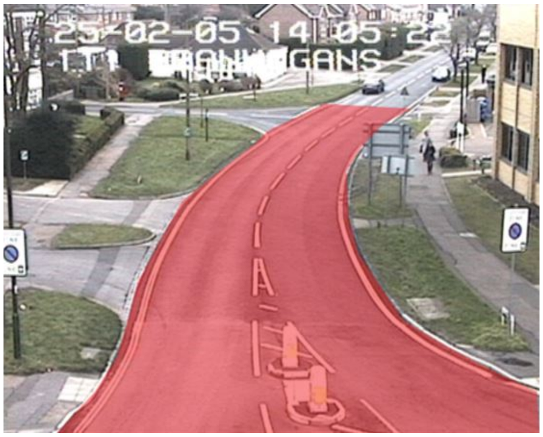

Fig. 2. i-LIDS Dataset No Parking zone

### *A. Gaussian Mixture Model (GMM)*

Segmentation of a moving foreground object in a video scene typically involves modeling the background without the moving object and then computing the error between the modeled background and the current sequence image. Among the different existing methods for background model construction and foreground extraction, Gaussian mixture model (GMM) has established itself as one of the most reliable and computationally efficient approaches. Originally introduced by [21], [22], the GMM approach models each pixel of an image as a mixture of Gaussian distributions. Intuitively, a pixel resulted from a single surface under a fixed illumination condition and sensor noise can be modeled using a single Gaussian distribution. Here, illumination changes over time could also be accommodated by utilizing an adaptive Gaussian. However, in real-world scenarios, a single pixel color could be contributed by multiple surfaces present in the image scene, along with the illumination changes over time. As a result, to model a scenario like this, the GMM method utilizes a mixture of adaptive Gaussian distributions. Foreground and background pixels are then differentiated based on the variance and persistence of each Gaussian in the mixture.

In a Gaussian mixture model, values observed for a particular pixel over time is treated as a “pixel process”. In practice, a “pixel process” incorporates a time series of how the pixel value (gray-scale or R, G, B values) changed over time. Hence, at any given point of time $t$, a pixel $(x, y)$ is represented by its history:

$$\{X_1, ..., X_t\} = \{I(x, y, i) : 1 \leq i \leq t\} \tag{1}$$

Here, $I$ is the gray-scale or R, G, B values of the pixel $(x, y)$ at time $i$. Each pixel value in an image scene corresponds to the surface radiance of the closest object encountered by the optical axis of the pixel toward the direction of the camera. Under the presence of a fixed background and illumination condition, the changes in the pixel value over time would be negligible. Hence, with the assumption that independent Gaussian noise was introduced through image acquisition and sampling, its density could be represented as a single Gaussian distribution with the mean pixel value as the center of the distribution. However, video sequences captured under uncontrolled environment would contain lighting changes along with background and foreground changes. Hence, any background subtraction-based moving object detection technique needs to track lighting changes in the scene. In addition, the constructed background model should also account for newly added background objects in the scene that was not present in the original model. However, this scenario poses a challenge for methods that require the newly added background object to be present longer than the previous object to be considered as part of the background. During this time window, the new object is typically considered as foreground and thus it introduces unexpected or poor tracking results. Based on these observations, Stauffer and Grimson in [21] argued that more recent observations should be emphasized when estimating the parameters of the Gaussian distributions.

Another important aspect related to background modeling is determining the characteristics of moving objects and stationary objects (background) in the scene. According to Stauffer and Grimson in [21] and [22], it is safe to assume that any moving object, even with relatively uniform surface area would exhibit more variance in terms of pixel values than stationary objects. On the other hand, stationary objects or background would provide consistent observations over time due to their repetitions.

All these observations contribute to how the Gaussian mixture model (GMM) creates and updates the background model. First, the recent history of each pixel (Eq. 1) is modeled using a mixture of $K$ Gaussian distributions. For a particular pixel $(x, y)$, the probability that it takes the current value $X_t$ can be defined as:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \times \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2}$$

Here, $K$ represents the number of Gaussian distributions in the mixture, $w_{i,t}$ is a weight factor for the $i$-th Gaussian at time $t$, and $\mu_{i,t}$ and $\Sigma_{i,t}$ represent the mean and covariance matrix of that Gaussian. Lastly, $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ represents the probability density for the current Gaussian. The value of $K$ is typically set between 3 and 5, depending on the availability of resources such as memory and computational power.

Once a mixture of Gaussian is computed for a particular pixel, every new observation for that pixel is matched against the mixture in order to find a distribution within a threshold standard deviation of 2.5. If a match is found for the new pixel, the parameters of the corresponding Gaussian distribution are updated. However, if no match is found, then a search is conducted for the least probable component of the corresponding mixture, which is then replaced by a new Gaussian distribution, modeling the newly observed pixel value. Stauffer and Grimson in [21], [22] treated the prior weights for each of the Gaussian distribution in the mixture as:

$$w_{k,t} = (1 - \alpha) w_{k,t-1} + \alpha(M_{k,t}) \tag{3}$$

Here, $\alpha$ represents the learning rate of the model, while the value of $M_{k,t}$ could be either 1 (for the distribution for which a match was found) or 0 (all other distributions). The parameters mean $\mu$ and standard deviation $\sigma$ for any unmatched distribution remain the same for the newly observed pixel value. On the other hand, the parameters of the Gaussian for which a match was found is updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{4}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \tag{5}$$

Here, $\mu_{t-1}$ represents the previous mean of the distribution, while $\sigma_{t-1}^2$ represents the variance. A second learning rate $\rho$ is also introduced here, which can be defined as:

$$\rho = \alpha\eta(X_t | \mu_k, \sigma_k) \tag{6}$$

The last step involves determining which Gaussian distributions in the obtained mixture represent the background model. Intuitively, appearance of a moving object in the image scene would result in pixel values that either do not match any of the existing distributions in the mixture or cause high variations in existing distributions. Hence, only the distributions with the least variation and high repetitions should be selected as background processes. To achieve this, the distributions in the mixture are ranked based on the $w/\sigma$ value to obtain a measure of relevance, which is then utilized to select the top B distributions as follows:

$$B = argmin_b(\sum_{k=1}^{b} w_k > T) \tag{7}$$

Here, $T$ is a threshold that represents what portion of the data needs to be considered for constructing the background model.

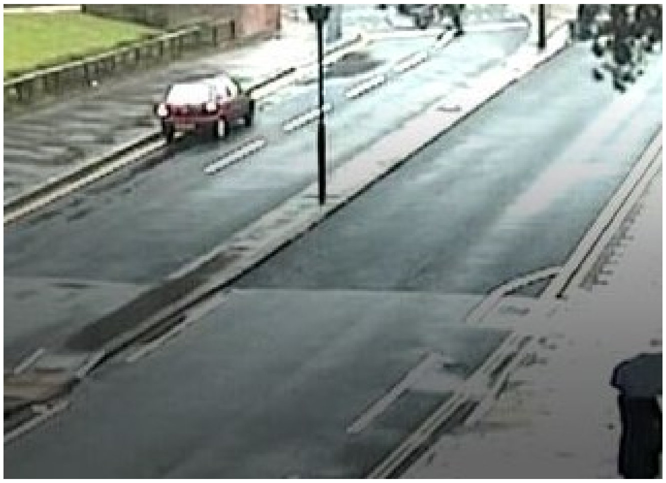

Fig. 3. i-lids dataset video input

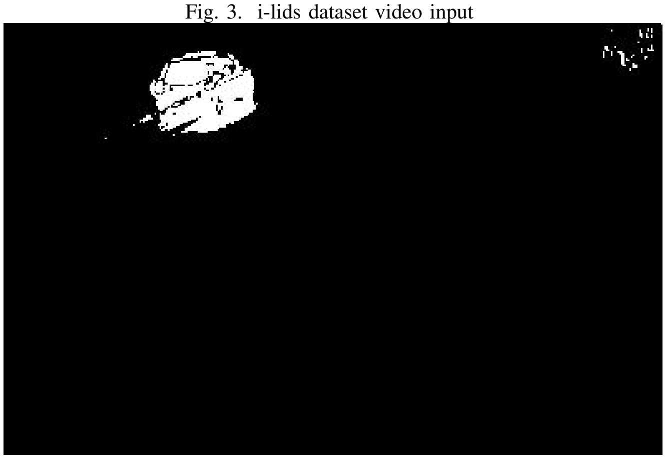

Fig. 4. Foreground Detection

## B. Kalman Filter for Motion Tracking

Once the object of interest is detected in an image scene, the next step involves tracking the motion of that object over time. Among the different motion tracking methods found in literature, Kalman filter [23] is one of the most well-known techniques. Also referred as linear quadratic estimation (LQE), the Kalman filter provides methodologies to estimate the state of a linear process using a recursive approach, with an objective function that minimizes the mean squared error. Due to the low computational complexity and the ability to provide optimal solution to any 1-D linear system under the presence of Gaussian noise [24], Kalman filter has been used in a wide range of applications over the last 50 years, which include motion tracking, global positioning and navigation systems, multi-modal information fusion, etc. [25].

In Kalman filtering-based motion tracking [26], the location of an object that we want to track in the image scene is first represented a series of states (history information). Then, any variation in the state over time can be modeled using a linear system:

$$X_t = A_t X_{t-1} + \eta \tag{8}$$

Here, $\eta$ represents a Gaussian white noise component having a covariance matrix $\Sigma^{\eta}$. Now, the state and the measurement can be related through the following measurement equation:

$$Y_t = D_t X_t + \epsilon \tag{9}$$

Here, $D_t$ represents a measurement matrix, while $\epsilon$ corresponds to another independent white noise component having a covariance matrix $\Sigma^{\epsilon}$. The objective here is to estimate the current state $X_t$ by utilizing the state history information. In other words, given a series of measurements over time represented as $(Y_1, Y_2, ..., Y_t)$, we need to find the probability density function $p(X_t|Y_1, ..., Y_t)$.

In theory, Kalman filter utilizes a recursive Bayesian filter to estimate an optimal solution in two stages: i) the prediction step and ii) the correction step. The prediction step involves finding the prior probability distribution using dynamic equation and the probability density function computed for all previously observed states stored in the state history. In the correction step, the posterior probability distribution is estimated using a likelihood function.

Originally, the Kalman filter was introduced to provide a methodology for estimating the current state of a linear system where the states follow a Gaussian distribution. In the prediction step, current state of a given variable or set of variables is estimated using the following state model:

$$X_t^p = A X_{t-1} \tag{10}$$

$$\Sigma_t^p = A \Sigma_{t-1} A^T + \Sigma_t^{\eta} \tag{11}$$

Here, $X_t^p$ represents the predicted state, while $\Sigma_t^p$ represents the predicted covariance matrix at a given time $t$. Here, the purpose of $A_t$ is to provide a relationship between the states observed at time $t$ and $t-1$. Hence, it is called the state transition matrix. Once $X_t^p$ and $\Sigma_t^p$ have been estimated, the correction step involves utilizing them to obtain the current state of the variable(s):

$$X_t = X_t^p + G_t[Y_t - D_t X_t^p] \tag{12}$$

$$G_t = \Sigma_t^p D_t^T [D_t \Sigma_t^p D_t^T + \Sigma_t^{\epsilon}]^{-1} \tag{13}$$

$$\Sigma_t = \Sigma_t^p - G_t D_t \Sigma_t^p \tag{14}$$

Here, $G_t$ represents a gain component referred as Kalman gain. This component is used for information propagation from one state to the other. Another important aspect of Kalman filtering is that the estimated current state $X_t$ follows a Gaussian distribution. The above formulation can also be adopted for nonlinear systems, where Taylor series expansion is used to construct an extended Kalman filter. Here also, the underlying assumption that the states of the variable(s) follow a Gaussian distribution holds true.

## IV. Experimental Results and Evaluations

The algorithm has been tested on i-LIDS dataset and the proposed technique implemented in MATLAB R2016a with the following system specifications:

- Processor: Intel(R) Xeon(R) CPU E5 - 1650 v3 @ 3.50 GHz
- Memory (RAM) 16 GB

### A. i-LIDS Dataset

The Imagery Library for Intelligent Detection Systems (i-LIDS) is one of well known dataset in computer vision and video processing, i-LIDS dataset developed by the Centre for Applied Science and Technology (CAST) in partnership with the Centre for the Protection of National Infrastructure (CPNI) [27], provides an event detection for the following scenarios:

- Doorway surveillance.
- Abandoned baggage detection.
- Sterile zone monitoring.
- Parked vehicle detection with alarm events consisting of suspiciously parked vehicles in an urban setting.

### B. i-lids Ground Truth

i-LIDS dataset is supplied with an XML based index describing at a high level the content and alarm events present in each video file. Table I illustrates the i-LIDS illegal parking dataset ground truth.

Event Detection performance on a scenario is rated using a weighted harmonic mean of a systems recall and precision known as the F1 measure. where are a , b and c as following:

- (a) True positive alarms (TP) : System alarms in response to a genuine alarm event.
- (b) False positive alarms (FP) : System alarms without the presence a genuine alarm event.
- (c) False negative alarms (FN) : Genuine alarm events not resulting in a system alarm.

According to i-LIDS [27] ”Systems have ten seconds to report an alarm state after an alarm event begins in the evaluation footage. During this time multiple alarm reports will be disregarded; an alarm event is either detected or not. After this ten second window, any further alarms reported will be deemed false positives”, see Fig. 5 for i-lids event detection evaluation.

The recall (Detection rate): $r = a/(a+c)$ The precision (probability of an alarm being genuine):$p = a/(a+b)$

$$F_1 = \frac{(\alpha+1)rp}{r+\alpha p} \tag{15}$$

TABLE I
I-LIDS ILLEGAL PARKING DATASET GROUND TRUTH

| File Name | Duration | Time | Rain | Snow | Alarm Events |
|---|---|---|---|---|---|
| PVTEA101a | 01:21:29 | Day | YES | NO | 26 |
| PVTEA101b | 00:24:40 | Night | No | NO | 8 |
| PVTEA102a | 00:51:56 | Day | NO | NO | 18 |
| PVTEA102b | 00:42:07 | Day | Yes | NO | 16 |
| PVTEA103a | 01:03:54 | Day | NO | NO | 16 |
| PVTEA103b | 00:27:13 | Dusk | NO | NO | 7 |
| PVTEA104a | 00:32:11 | Night | NO | NO | 8 |
| PVTEA104b | 00:32:11 | Night | NO | NO | 9 |
| PVTEA201a | 00:18:50 | Dusk | NO | NO | 4 |
| PVTEA201b | 00:41:38 | Day | Yes | NO | 9 |
| PVTEA201c | 00:32:28 | Night | NO | NO | 12 |
| PVTEA201d | 00:29:05 | Day | NO | NO | 8 |
| PVTEA202a | 00:24:02 | Day | NO | NO | 5 |
| PVTEA202b | 01:05:23 | Day | Yes | NO | 17 |
| PVTEA301a | 00:15:29 | Day | NO | NO | 3 |
| PVTEA301b | 00:28:07 | Day | Yes | NO | 8 |
| PVTEA301c | 00:27:09 | Day | NO | Yes | 6 |
| PVTEA301d | 00:16:15 | Dusk | NO | NO | 3 |
| PVTEA301e | 00:20:27 | Day | NO | NO | 6 |
| PVTEN102d | 00:19:43 | Day | NO | NO | 1 |
| PVTEN201a | 00:29:56 | Day | Yes | NO | 2 |
| PVTEN201b | 00:30:00 | Day | NO | NO | 1 |
| PVTEN201c | 00:20:05 | Day | NO | NO | 2 |
| PVTEN201d | 00:19:54 | Day | Yes | NO | 2 |
| PVTEN202a | 00:19:54 | Night | NO | NO | 7 |
| PVTEN202b | 00:30:01 | Dusk | NO | NO | 4 |
| PVTEN202c | 00:30:00 | Day | NO | NO | 7 |
| PVTEN301b | 00:29:59 | Day | NO | NO | 2 |
| PVTEN301c | 00:19:58 | Day | Yes | NO | 2 |

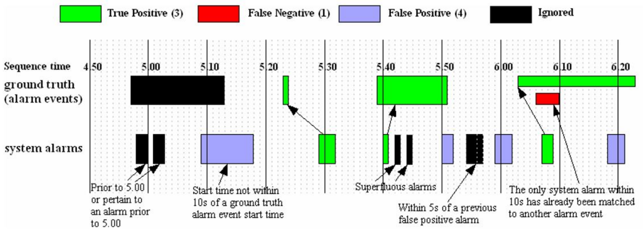
True Positive (3)
False Negative (1)
False Positive (4)
Ignored
Sequence time
4.50
5.00
5.10
5.20
5.30
5.40
5.50
6.00
6.10
6.20
ground truth (alarm events)
system alarms
Prior to 5.00 or pertain to an alarm prior to 5.00
Start time not within 10s of a ground truth alarm event start time
Superfluous alarms
Within 5s of a previous false positive alarm
The only system alarm within 10s has already been matched to another alarm event

Fig. 5. Ilids Event Detection evaluation

where $\alpha$ is the recall bias = 0.55.

The first scenario PVTEA101a.mov with a duration of 01:21:29 and it has 27 illegal parking events in raining weather condition, the result shows that 24 evens detected and 4 evens were missed detected, shown in Table II.

Table II illustrates the output result for PTEA101a. The system has detected 7 (TP) ”True positive alarms” and 11 events after 10 Seconds late from the started parking time, for example, in the PVTEA101a at 381 seconds, the system has alarmed at 352 seconds, so it missed 29 seconds compared to the ground truth. Moreover, the system has missed 4 evens for example, in the 968 seconds as shows in TableII. After the evaluate calculation, the recall (Detection rate) is 0.38, and the precision (P) is 0.388. According to F1 measure 15 the value of F1 is 39%, see Fig. 7 for the performance matrix.

The next video in the dataset is PVTEA101b with a duration of 24 minutes at night time and it has 8 alarms, see Table III.

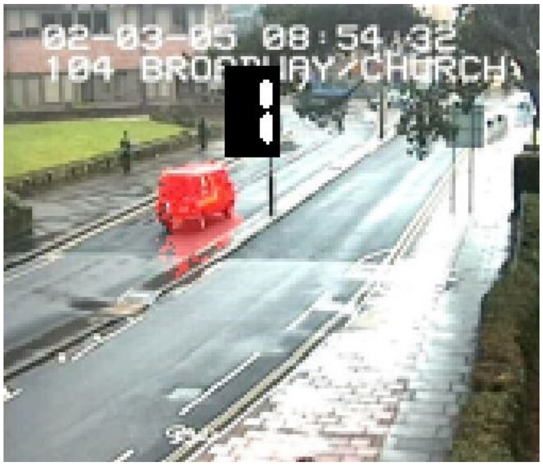

Fig. 6. The Proposed Vehicle Tracking System

TABLE II
THE TRACKING RESULTS FOR PVTEA101A COMPARED TO I-LIDS DATASET GROUND TRUTH

| File Name | Groundtruth | Detection | Error in (Sec) | Status |
|---|---|---|---|---|
| PVTEA101a | 381 | 352 | 29 | (FP) |
| PVTEA101a | N/A | 451 | (FP) | (FP) |
| PVTEA101a | 968 | 0 | (FN) | (FN) |
| PVTEA101a | N/A | 700 | (FP) | (FP) |
| PVTEA101a | N/A | 878 | (FP) | (FP) |
| PVTEA101a | 1055 | 0 | (FN) | (FN) |
| PVTEA101a | 1137 | 1083 | 54 | (FP) |
| PVTEA101a | 1138 | 0 | (FN) | (FN) |
| PVTEA101a | 1278 | 1280 | -2 | (TP) |
| PVTEA101a | 1370 | 1391 | -21 | (FP) |
| PVTEA101a | 1455 | 1455 | 0 | (TP) |
| PVTEA101a | 1643 | 1644 | -1 | (TP) |
| PVTEA101a | 1738 | 1739 | -1 | (TP) |
| PVTEA101a | 1874 | 1898 | -24 | (FP) |
| PVTEA101a | 2080 | 2081 | -1 | (TP) |
| PVTEA101a | 2316 | 2317 | -1 | (TP) |
| PVTEA101a | N/A | 2524 | (FP) | (FP) |
| PVTEA101a | 2616 | 2616 | 0 | (TP) |
| PVTEA101a | 2675 | 2691 | -16 | (FP) |
| PVTEA101a | 2905 | 2939 | -34 | (FP) |
| PVTEA101a | 3016 | 0 | (FN) | (FN) |
| PVTEA101a | 3024 | 3041 | -17 | (FP) |

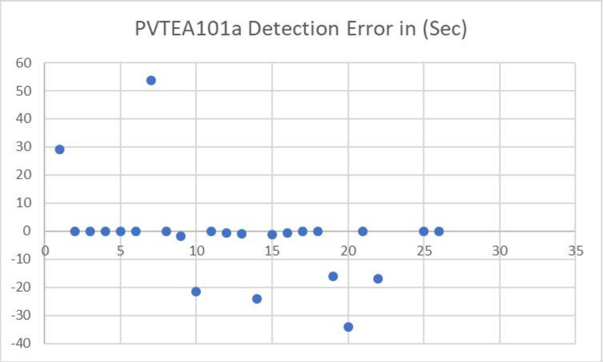

Fig. 7. PVTEA101a Performance Matrix

TABLE III
THE TRACKING RESULTS FOR PVTEA101B COMPARED TO I-LIDS DATASET GROUND TRUTH

| File Name | Groundtruth | Detection | Error in (Sec) | Status |
|---|---|---|---|---|
| PVTEA101b | 360 | 374 | -14 | (FP) |
| PVTEA101b | 500 | 503 | -3 | (TP) |
| PVTEA101b | 678 | 678 | 0 | (TP) |
| PVTEA101b | 779 | 780 | -1 | (TP) |
| PVTEA101b | 920 | 919 | 1 | (TP) |
| PVTEA101b | 1019 | 1021 | -2 | (TP) |
| PVTEA101b | 1259 | 1258 | 1 | (TP) |
| PVTEA101b | 1363 | 1366 | -3 | (TP) |

Table III shows the detection results for PTEA101b. The system has detected a 7 (TP) true positive alarms and 1 event, but 14 Seconds late after the event started, for example, in the PVTEA101b at 360 seconds, the system has given alarm at 374 seconds, which is 14 seconds late compared to the ground truth. Moreover, the system did not miss any detection as shows in table II . The recall (Detection Rate) is %875, and the precision (P) is 0.875.finally, F1 measure is 88%, see Fig. 8 for the performance matrix.

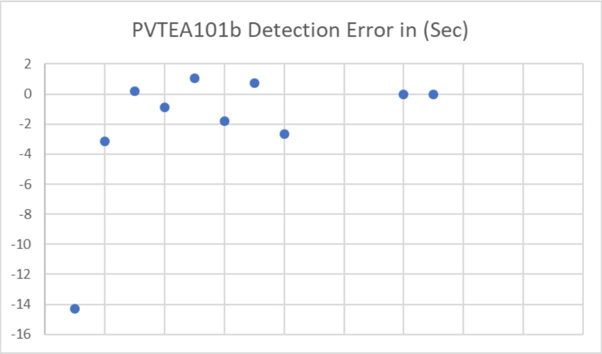

Fig. 8. PVTEA101b Performance Matrix

The third experiment to detect illegal parking using the video (PVTEA102a), according to the dataset i-LID, PVTEA102a is consider as (Level 1) with a duration of 00:51:56 at day time with some cloud, and it has a 26 illegal parking evens. The result illustrated in Table IV.

The system in PVTEA102a has detected a 7 (TP) within the maximum 10 seconds late according to the ground truth. On the other hand, 14 evens have been detected but after 10 seconds, hence they are considered as (FP) false positive. Moreover, the system did not detect 4 illegal parking evens. The recall (Detection Rate) 0.38, and the precision (P) is 0.33. According to f1 measure in eq.15 the value of F1 is 35%. The performance matrix shown in Fig. 9.

Table V illustrates the output result for PTEA102b, the system has detected a 4 TP ”True positive within the maximum of 10 seconds late of the detection. Moreover, the system detected 15 (FP) False positives beside an additional of 7 missed detections as false negative . According to the evaluation formula, F1 measure is 23%.

TABLE IV
THE TRACKING RESULTS FOR PVTEA102A COMPARED TO I-LIDS DATASET GROUND TRUTH

| File Name | Groundtruth | Detection | Error in (Sec) | Status |
|---|---|---|---|---|
| PVTEA102a | 352 | 381.16 | -29 | (FP) |
| PVTEA102a | 451 | | (FN) | (FN) |
| PVTEA102a | 700 | | (FN) | (FN) |
| PVTEA102a | 878 | 967.8 | -90 | (FP) |
| PVTEA102a | N/A | 1054.56 | (FP) | (FP) |
| PVTEA102a | N/A | | (FP) | (FP) |
| PVTEA102a | N/A | | (FP) | (FP) |
| PVTEA102a | 1083 | 1136.68 | -54 | (FP) |
| PVTEA102a | N/A | 1137.72 | (FP) | (FP) |
| PVTEA102a | 1280 | 1278.16 | 2 | (TP) |
| PVTEA102a | N/A | 1369.64 | (FP) | (FP) |
| PVTEA102a | 1391 | | (FN) | (FN) |
| PVTEA102a | 1455 | 1454.96 | 0 | (TP) |
| PVTEA102a | 1644 | 1643.28 | 1 | (TP) |
| PVTEA102a | 1739 | 1738.24 | 1 | (TP) |
| PVTEA102a | N/A | 1873.92 | (FP) | (FP) |
| PVTEA102a | 1898 | 1891.72 | 6 | (TP) |
| PVTEA102a | 2081 | 2079.8 | 1 | (TP) |
| PVTEA102a | N/A | 2292.68 | (FP) | (FP) |
| PVTEA102a | 2317 | 2316.48 | 1 | (TP) |
| PVTEA102a | 2524 | 2615.88 | -92 | (FP) |
| PVTEA102a | 2616 | 2674.92 | -59 | (FP) |
| PVTEA102a | 2691 | | (FN) | (FN) |
| PVTEA102a | 2939 | 2904.84 | 34 | (FP) |
| PVTEA102a | N/A | 3016.32 | (FP) | (FP) |
| PVTEA102a | 3041 | 3023.96 | 17 | (FP) |

TABLE V
THE TRACKING RESULTS FOR PVTEA102B COMPARED TO I-LIDS DATASET GROUND TRUTH

| File Name | Groundtruth | Detection | Error in (Sec) | Status |
|---|---|---|---|---|
| PVTEA102b | 509 | 504.36 | 5 | (TP) |
| PVTEA102b | 664 | | (FN) | (FN) |
| PVTEA102b | N/A | 760.08 | (FP) | FP |
| PVTEA102b | N/A | 775.52 | (FP) | (FP) |
| PVTEA102b | 785 | 831.2 | -46 | (FP) |
| PVTEA102b | 864 | 878.32 | -14 | (FP) |
| PVTEA102b | N/A | 935.88 | (FP) | (FP) |
| PVTEA102b | N/A | 987.24 | (FP) | (FP) |
| PVTEA102b | 1099 | 1105.28 | -6 | (TP) |
| PVTEA102b | 1204 | | (FN) | (FN) |
| PVTEA102b | 1377 | | (FN) | (FN) |
| PVTEA102b | 1590 | 1597.72 | -8 | (TP) |
| PVTEA102b | 1630 | | (FN) | (FN) |
| PVTEA102b | 1630 | 1737.12 | -107 | (FP) |
| PVTEA102b | 1836 | 1891.48 | -55 | (FP) |
| PVTEA102b | N/A | 2005.24 | (FP) | (FP) |
| PVTEA102b | 2059 | 2085.12 | -26 | (FP) |
| PVTEA102b | N/A | 2106.28 | (FP) | (FP) |
| PVTEA102b | 2260 | 2258.16 | 2 | (TP) |
| PVTEA102b | N/A | 2331.08 | (FP) | (FP) |
| PVTEA102b | N/A | 2356.64 | (FP) | (FP) |
| PVTEA102b | N/A | 2359.32 | (FP) | (FP) |
| PVTEA102b | N/A | | (FP) | (FP) |

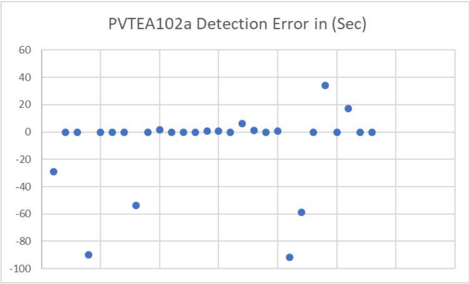

Fig. 9. PVTEA102a Performance Matrix

## V. CONCLUSION

In this paper, we presented a tracking technique to detect illegal parking using i-LIDS dataset. The results shown various performance of f1 measure. The worst performances can be observed in PVTEA102b due to the raining weather. During processing the video, the sun (light source) appears several times, which causes different reflected light, resulting in lower performance of 22% compared the to stable weather condition in PVTEA101b with a duration of 00:24:40 at night time, and it has a performance of 87%. The low detection and tracking performance of PVTEA102b PVTEA101a and PVTEA102a is due to their weather condition, which causes reduced of the light source. This problem can be dealt by applying Physics based Image formation models such as illumination estimation and surface spectral reflectance estimation would be be applied as future work to tackle the illumination challenge in order to get better performance and accuracy.

## REFERENCES

[1] Department for Transport of United Kingdom, "Facts on road fatalities," website, 2015. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_-data/file/448037/road-fatalities-2013-data.pdf

[2] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[3] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, pp. 359–381, 2003.

[4] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[5] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, *Handbook of Unmanned Aerial Vehicles*. Springer, ch. Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring, pp. 2643–2666.

[6] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2011.

[7] N. E. Buch, "Classification of vehicles for urban traffic scenes," Ph.D. dissertation, Kingston University, 2010.

[8] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, 2002.

[9] B. Morris and M. Trivedi, "Improved vehicle classification in long traffic video by cooperating tracker and classifier modules," in *IEEE International Conference on Video and Signal Based Surveillance*, no. 9-9, 2006.

[10] D. Bloisi and L. Iocchi, "Argosa video surveillance system for boat traffic monitoring in venice," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 7, pp. 1–23, 2009.

[11] A. Opelt, A. Pinz, and A. Zisserman, "A boundary-fragment-model for object detection," in *European Conference on Computer Vision*, 2006, pp. 575–588.

[12] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1083–1698, 2008.

[13] K. Park, D. Lee, and Y. Park, "Video-based detection of street-parking violation," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2007.

[14] P. Nguyen and H. Le, "Multi-modal particle filter based motorcycle tracking system," in *Trends in Artificial Intelligence*, ser. LNCS, vol. 5351, 2008, pp. 819–828.

[15] C. Huang and W. Liao, "A vision-based vehicle identification system," in *International Conference On Pattern Recognition*, vol. 4, 2004, pp. 364–367.

[16] N. Kanhere and S. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 148–160, 2008.

[17] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *IEEE Intelligent Transportation Systems Conference*, 2006.

[18] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transportation Research Record*, vol. 1944, pp. 82–88, 2005.

[19] S. Messelodi, C. Modena, N. Segata, and M. Zanin, "A kalman filter based background updating algorithm robust to sharp illumination changes," in *International Conference on Image Analysis and Processing*, ser. LNCS, vol. 3617, 2005, pp. 163–170.

[20] T. Gao, Z. Liu, W. Gao, and J. Zhang, "A robust technique for background subtraction in traffic video," *Advances in Neuro-Information Processing*, vol. LNCS 5507, pp. 736–744, 2009.

[21] C. Stauffer and W. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 246–252.

[22] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[24] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. New York: Dover, 2005.

[25] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[26] F. Porikli and A. Yilmaz, *Video Analytics for Business Intelligence*, C. Shan, F. Porikli, T. Xiang, and S. Gong, Eds. Springer, 2012.

[27] Home Office OF UK , "Imagery library for intelligent detection systems," website, 2014. [Online]. Available: https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems