Real-Time Search-Free Multiple License Plate Recognition via Likelihood Estimation of Saliency

Amin Safaei, Hongying L. Tang, and Saeid Sanei

Abstract

In this paper, we propose a novel search-free localization method based on 3-D Bayesian saliency estimation. This method uses a new 3-D object tracking algorithm which includes: object detection, shadow detection and removal, and object recognition based on Bayesian methods. The algorithm is tested over three image datasets with different levels of complexities, and the results are compared with those of benchmark methods in terms of speed and accuracy. Unlike most search-based license-plate extraction methods, our proposed 3-D Bayesian saliency algorithm has lower execution time (less than 60 ms), more accuracy , and it is a search-free algorithm which works in noisy backgrounds.

Index Terms

Automatic license plate recognition, Bayesian, intelligent transportation system, localization, object tracking, saliency map detection, shadow removal.

I. INTRODUCTION

Nowadays, intelligent transportation systems (ITS) have a wide impact on people's daily activities. Advanced computer vision technologies and automatic license plate recognition (ALPR) systems are used at the core of ITS infrastructure to identify the vehicles licenses. These technologies are required in many applications such as red-light enforcement, speed enforcement, electronic payment systems and traffic surveillance. ALPR process usually consists of four main steps: image acquisition, localization, segmentation, and optical character recognition.

A. Safaei, Hongying L. Tang and S. Sanei are with the Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, UK.

ALPR systems have to cope with challenges such as images taken under different environmental conditions and lightings, sizes, and orientations. The plate numbers may be occluded or have their locations varied. In real-time operations e.g. ITS, ALPR process has to operate fast and accurately. In specific applications, even a single late detection or mis-detection of a moving car through the camera scene may not be tolerable. As such, the problem of improving the performance and accuracy of ALPR systems has become a crucial research area in ITS applications.

Localization is the most challenging step of any ALPR process and is strongly influenced by the accuracy and execution time of the overall system. Several methods have been suggested in the literature for license plate (LP) localization. According to Du et al. [1], since an LP can be distinguished by its contextual features, there is no need to process the entire scene to find the plate number. On the other hand, Anagnostopoulos et al. [2] argued that global image information techniques such as connected component analysis (CCA) can be used. CCA works for both grey level and binary images. This technique scans the image and finds those pixels in a connected component that share similar intensity values.

Matas et al. [3] suggested a region-based approach to extract character-like regions. The plate is assumed to be a linear combination of these small regions. Although this method is time-consuming, it is robust to rotation and has a high accuracy.

Broumandnia et al. [4] suggested another localization method called partial image analysis where the image is scanned row by row and the number of edges within each N rows is counted. This method is very fast and simple to implement, however it depends on the plate size and its distance from the camera.

Zunino et al. [5] proposed a method based on hierarchical representation using vector quantization (VQ). Using VQ different (but lower) image resolutions can be achieved where a vector of close pixels can be represented using only one value (such as their cluster centre). Therefore, the regions of interest such as those of LP can be better recognized.

Statistical measurement methods have also been used in LP localization. Lee et al. [6] proposed a method to estimate the LP region by identifying the blocks with high edge magnitudes or high edge variances. Although Lee et al. claim an accuracy of 92.5%, other works such as that proposed by Anagnostopoulos et al. [2], argue that not all the blocks detected by this algorithm are the actual LPs. Probabilistic object tracking in videos introduced by Yalcin et al. [7] is a method for LP location estimation and car tracking in successive video frames. In this method, probability density propagation is used to estimate the object and filter the measurements. According to Ying et al. [8], the algorithm developed by Yalcin et al. can only improve the computation time and not the accuracy.

Most algorithms used in LP localization, such as by Du et al. [1], are based on the presumption that an LP can be viewed as irregularities in the image texture. Rapid changes in the local characteristics of

the car image point to the LP location. Although this algorithm is simple to implement and has a high localization rate, the search process is time-consuming and several parameters need to be adjusted.

Search-based approaches often employ a sliding window to scan and search the entire image, which results in high recognition rate. However, they are computationally slow. On the other hand, the methods based on CCA are rapid but require an accurate binarisation step and an accurate and effective threshold selection.

In our earlier work [9, 10], we presented a robust search-free LP localization, incorporating hierarchical saliency and a novel search-free method for accurately locating the LP based on the estimation of saliency, local variance and the use of Gabor functions to ascertain the choice of the candidate LP. In comparison with the other methods such as those proposed by Lee et al. [6], Anagnostopoulos et al. [2], Yalcin et al. [7] and Isard et al. [8], this method performs faster and more reliably as it avoids any exhaustive search. For the original saliency map detection (SMD) method introduced by Itti et al. [11], however, further modifications are required to enhance the speed.

One main drawback of most existing algorithms is that they attempt to perform the localization of the vehicles using still images without taking into account the vehicle movement. As a result, they mistakenly detect undesired objects such as trees, road signs, or shadows as salient regions. Therefore exploiting the motion in videos by tracking algorithms can lead to a more efficient ALPR system.

In the last few decades, many researchers have presented the object detection and tracking algorithms. The algorithm presented by Shantaiya et al. [12] uses Kalman filter and optical flow. This method has a high performance in scenarios where detecting moving object in similar background. However is not able to track low resolution object. Kodama et al. [13] uses particle filter and optical flow for object tracking. Similar to [12], this method requires large computation time and is not efficient for tracking objects in low resolution. On the other hand Li et al. [14] uses background cues for tracking objects. Although this algorithm has a high accuracy, it is sensitive to occlusion and illumination changes. A novel tracking algorithm presented by Aslani et al. [15] uses blob analysis for object segmentation and optical flow field vectors for object tracking in traffic surveillance. This method is a more efficient comparing to [12, 13, 14]. Though experiments shows that this algorithm is sensitive to camera motion and only works for tracking single objects.

In Bhaskar et al. [16], a unique algorithm for vehicle data recognition and tracking using Gaussian mixture model and blob detection methods is proposed.

Wang et al. [17] proposed an effective real-time background extraction and moving object detection method with less memory usage comparing to the methods such as Chiu et al. [18]. Jiang et al. [19] proposed another background extraction algorithm, i.e. Partition Weighed Histogram (PWH). The

algorithm assumes that the background does not vary considerably in a small neighborhood over time.

Russell et al. [20] also proposed a simple method for detecting and removing shadows in traffic videos by using spatial intensity relationship between pixels associated to a scanned image line. This method performs the classification on image lines to differentiate between the background and the shadow and has a high accuracy rate.

In designing a robust tracking algorithm, the moving shadows cause major problems. Yuan et al. [21] proposed a shadow detection technique with a surface descriptor, named colour shade. This algorithm has high accuracy and can remove shadows from the images and reconstruct the image without shadows. However, the algorithm is time-consuming, since it uses the gradients of the images and the derivations in two directions. Tian et al. [22] discussed and evaluated a number of shadow detection algorithms. One important and conventional method for scene shadow detection uses texture analysis. In this method, in order to distinguish between shadows and moving objects, texture features are analysed to distinguish the moving regions and the background frames. Both structural and statistical approaches have been used for image texture analysis.

Haralick et al. [23] proposed fourteen statistical features to describe a texture. Authors in [23] believe that texture is an important characteristic for identifying the objects or ROIs in an image and they describe a method to estimate texture features based on grey tone spatial dependencies. A grey level co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values with a given offset. In [23], two kinds of decision rules are proposed namely the piecewise linear and the min-max rules. To extract texture features, this method obtains four features of energy, entropy, contrast, and inverse difference moment. The method calculates these features for the previous and the current frames. Then, feature similarity between the current and the previous frames in the moving region is calculated. Finally, by comparing the texture of moving regions to the background, the region of interest is estimated.

Tracking in complex environments using Bayesian method has always been an interesting research area. Xiang [24] argue that such an approach has high accuracy and is computationally efficient in real-time speed completely. On the other hand Jamieson et al. [25] proposed a Bayesian estimation method to sequentially update the probability of detection for tracking. A beta distribution is used for the prior, which can be centred on the best a priori guess for the probability of detection. Stone et al. [26] proposed many tracking algorithm based on Bayesian method, such as likelihood TBD (track before detection), which is implemented as the likelihood ratio TBD. Although use of Bayesian method for tracking is first introduced in this paper. However some research have used Bayesian method for tracking LPs, e.g. Zhang et al. [27] proposed a new visual object tracking algorithm to handle complex scenarios with

high performance using a joint Bayesian and Kalman filter (BKF) approach. Also, Mazurek et al. [28] proposed a tracking system for vehicles moving away from the camera. This algorithm uses a block matching technique and Bayesian decision making for tracking.

After object detection, localization step is to find the coordination of LP in each object, detected in the tracking process. Reji et al. [29] presented a comprehensive review of LP localization methods in term of detection rate, accuracy, merits, demerits, computational power and execution time. According to this review, Roomi et al. [30] and Wazalwar et al. [31] have highest success rate. Roomi et al. [30] method use LP region rough detection and region of interest estimated by vertical gradients and use of 2D discrete wavelet transform. On the other hand, Wazalwar et al. [31] make use of a smoothing operation, detection of edges and dilation operation in their localization algorithm.

This paper is inspired by [2, 11, 14, 20, 31] and we extend our previous work [9, 10] combined with Bayesian approach used in [27, 28] to estimate the 3-D salient regions by exploiting the motion information in the videos. The proposed algorithm is fast and robust when used in the changing environment and is suitable for localizing multiple plates.

The rest of the paper is organized as follows: In Section II, the capability of proposed method for object detection, shadow detection, object recognition and adaptive localization is investigated, whereas Section III demonstrates the experimental and numerical results. Section IV concludes the paper.

II. The Proposed Method

In this section, we propose a technique to exploit multi-frame images (videos). In definition of a new 3-D salient region, the shadows of the moving objects are detected and removed, and the car object is then recognized. Finally, a new adaptive localization technique is introduced, which makes the algorithm robust for real-time applications. In each step, the related work is discussed and evaluated.

A. 3-D Saliency

Saliency refers to a discriminatory identity based on some properties of an object or the regions of an image to be distinguished from their background or surroundings. Although Itti et al. [11] claim a high localization rate and speed, compared to the conventional methods, the speed of their algorithm requires to be further increased to suit an ITS application.

Thus, we propose a 3-D saliency method to speed-up the Itti's SMD method by detecting the moving objects as salient regions. In this method, a region is salient if something particular happens inside or around it and therefore, the objects without movement are not counted even if they have different
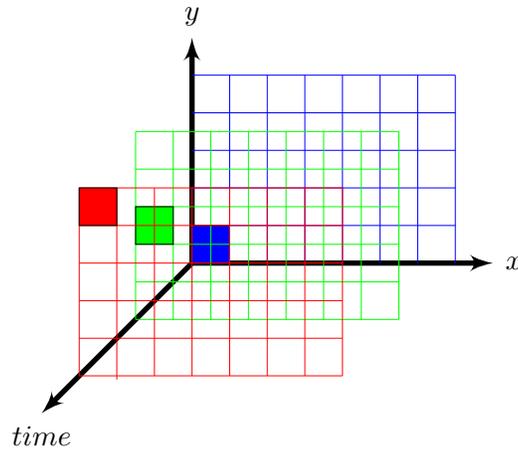
Figure 1: 3-D Saliency method detects the regions of moving objects as salient regions.

intensities, colours, or orientations related to their surroundings. A symbolic presentation of the concept is given in Figure 1. In the following sub-section, our proposed tracking algorithm is discussed.

B. Tracking

The tracking step determines the regions including vehicles in each image and finds relations between these regions in the video sequences. Adding this step to the localization process of ALPR system results in the reduction of the detection error and an increase in the speed. The proposed algorithm involves:

- Object Detection: locating initial region of the vehicles by subtracting the current image from the image background.
- Shadow Detection: eliminating the shadows of the vehicles, which limits the salient region and allows the separation of regions by their shadows.
- Object Recognition: identifying the relations between the detected objects in the current image and those in the previous images of the sequence.

Each of these steps are discussed in the following subsections.

1) Object Detection: Object detection is the first step of the vehicle tracking process. Object detection is defined as finding the coordination of each object in current image using the information of current and previous image frames.

In our proposed algorithm the first step in object detection results in removing the background.

Unlike [17] where the colour images are used, in our algorithm, the processing time is reduced when grey images are used.
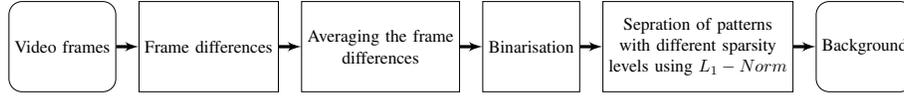
Figure 2: Block diagram of the proposed background extraction method.

Figure 2 shows our proposed background extraction method. As shown in this figure, our algorithm extracts the background from the input images. The input images have in them some objects that move, so similar to [16] an iterative (frame-by-frame) algorithm is proposed to extract the background and update it in the successive frames.

Comparing our proposed method with the one in [19], we noticed that although this algorithm has a good performance in the low traffic flow, it appears to be very time-consuming and exhibits low accuracy in medium and high traffic flow. On the other hand, the extracted background in [17] is fixed, whereas in our algorithm, the pixels values can be updated.

When the background is extracted, the object detection operation can be performed. The block diagram of proposed method for object detection is shown in Figure 3.

At the beginning of object detection algorithm, the difference image is compared with a threshold to determine if there has been any considerable change in the pixel values.

$$I_{diff}(i,j) \geq TH \Longrightarrow I_{Bin}(i,j) = 1, \quad changed.$$
$$I_{diff}(i,j) < TH \Longrightarrow I_{Bin}(i,j) = 0, \quad unchanged.$$

(1)

where i, and j are the row and column indices, respectively. TH is the binarisation threshold that can be assigned using Otsu's method [5].

Then, $L_1 - Norm$ is performed to calculate the sum of neighborhood pixels.

$$\|I(m,n)\|_1 = \sum_{i=m-M}^{m+M} \sum_{j=n-N}^{n+N} |I(i,j)|$$

(2)

where M and N denote the size of window around the current pixel (n,m). In the next step, the binary image is projected on the vertical direction and the columns with high numbers of foreground pixels are determined as regions of interest (ROI). Finally, using CCA, the objects are detected.
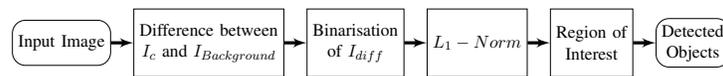
Figure 3: Block diagram of the object detection algorithm.

Comparing to our proposed object detection method, [17] proposes an algorithm of object segmentation based on background subtraction. In this method, the difference between the desired region and the background is calculated for both luminance and chrominance (signals used in video systems to express the colour information of the image). A pixel is then labeled as object if each of the differences has a value larger than the threshold. Although this method exhibits high accuracy because of incorporating the variations in two dimensions, it has some disadvantages such as:

- the need for extra computation cost of the luminance and chrominance components,
- it requires the input image to be colour,
- it demands at least two times additional computation time, since the differentiation is calculated in two dimensions.

In our proposed object detection method, the coordination of each object using the information of current and previous image frames is used, and similar to [17, 18, 19], the background extraction algorithm is performed. For images with dark shadows and irregularities in the shape, tracking algorithms such as [12, 13, 14, 15] could also detect the objects. However these algorithms have their own challenges such as: not tracking objects with variation in movement, some methods require large computation time, not efficient for tracking objects in low resolution and some are sensitive to occlusion and illumination changes.

2) Shadow Detection: In the development of robust dynamic-scene analysis algorithms, the moving shadows cause major problems. Since the shadow regions are often misclassified as objects, they may cause errors in the tracking step of the localization. In daylight conditions, shadows are more difficult to detect and sometimes detected as cars. This dramatically affects the localization rate. Shadows therefore impose irregularities in the shape of the detected regions or objects.

Using the algorithm in [23] for extraction of texture feature, we can obtain four features to describe the texture of image. Let Eng, Ent, Con and Inv denote the features of energy, entropy, contrast and inverse difference moment respectively. The description of the algorithm is given below:

- To calculate four features of background frame and current frame respectively. Let use subscript C to denote the features of current frame and subscript B the background frame.
- To calculate the feature similarity between the current frame and the background frame in moving regions. $D(i, j)$ determines whether the pixel is related to foreground or background.

$$for \ all \ i,j$$

$$if \ D(i,j) = 1, \ then$$

$$NUM = Eng_C \times Eng_B + Ent_C \times Ent_B \ +$$

$$Con_C \times Con_B + Inv_C \times Inv_B$$

$$DEN = \sqrt{Eng_C^2 + Ent_C^2 + Con_C^2 + Inv_C^2} \ \times$$

$$\sqrt{Eng_B^2 + Ent_B^2 + Con_B^2 + Inv_B^2}$$

$$Sim = \frac{NUM}{DEN}$$

$$end \ if$$

$$if \ Sim \geq TH, then$$

$$(shadow)D(i,j) = 0$$

$$end \ if$$

$$end \ for$$

where:

- Eng_C= energy of the current frame.
- Eng_B= energy of the background.
- Ent_C= entropy of the current frame
- Ent_B= entropy of the background.
- Con_C= contrast of the current frame
- Con_B= contrast of the background.
- Inv_C= inverse of the current frame
- Inv_B= inverse of the background.
- $D(i,j)$= the pixel at the position (i,j) related to foreground or background.

If $Sim \geq TH$, we can conclude that the texture of the current frame is similar to that of the background frame and also according to Luo et al. [32], we can exclude the pixels from the moving regions .

The methods described in [23, 32], based on our experimental analysis fail to distinguish black objects from the shadows. In our image datasets, the shadows can be extremely dark and, consequently, the background texture is completely removed.

| Detected objects | → | Detecting, connecting and eliminating white objects | → | Detecting and eliminating black objects | → | Eliminating objects similar to background | → | Shadow detection and removel |

Figure 4: Block diagram of the shadow detection algorithm.

The block diagram of our proposed shadow detection method is illustrated in Figure 4. In this method, the detection is performed in three steps. In the first step, white objects are recognized. Then, the detected white objects are removed from the difference image and black objects are located. Finally, black objects which are very close to white objects are removed as shadows of white objects. As shown in Figure 4, the detected objects with their shadows are processed by the algorithm and the white objects are detected. According to Zhu et al. [33], in order to detect white objects, using background image and the current image, the optical gain matrix (OGM) D can be defined as:

$$D(i,j) = \frac{I_{Curr}(i,j)}{I_{Back}(i,j)} \tag{4}$$

where I_{Curr} is the current image and I_{Back} is the background image. Parameters $[T_1, T_2]$ are determined to detect the shadow region using the following equation:

$$I_{Shadow} = \begin{cases} 1 & T_1 \leq D(i,j) \leq T_2 \\ 0 & else \end{cases} \tag{5}$$

where T_1 and T_2 are empirically set to 0.01 and 1.05 respectively. When removing shadow pixels from the detected objects, large white objects are found using the connected component technique.

According to Figure 4, connecting the white objects is the next step of our proposed shadow detection algorithm. In this step, based on the minimum distance among objects, the algorithm determines whether they are connected together or not. Then, we remove the pixels belonging to the detected object and then we detect the black objects. In this step, to determine the shadows of black objects, Equation (5) is used by setting T_1 and T_2 respectively to 0.5 and 1.01.

Black objects which are close to white objects are considered as shadows and are then eliminated. Afterwards, all the remaining white and black objects are compared to the background. Some objects which are very similar to the background are also removed.

To evaluate our proposed shadow removal method, we compared the results to the textural features analysis and shadow detection in Luo et al. [32]. We observed that when the shadows are strong, the

Approach	Accuracy	Computation time (ms)
Textural features	47.3%	162.7
Proposed algorithm	78.6%	12.5

Table I: Comparison of our proposed shadow detection method to [32].

background texture disappears and the algorithm cannot distinguish between shadows and black objects. This problem leads to low accuracy in the analysis of videos with high luminance. Table I shows the comparison between the shadow detection by the textural feature algorithm in [32] and our proposed shadow removal method.

Superiorities of our proposed shadow detection algorithm are as follows: unlike other state of the art methods that fail to distinguish black objects from the shadows, our method perfectly removes shadows from high grey-level objects; for accurately determine correct connected object and not detecting dark object as a shadow, black objects which are close to white objects are considered as shadows and computation time is low. Also, the grey shadows of black objects can be removed. However, this is not the case in many black objects.

3) Object Recognition: An intelligent ALPR system has to detect and track potential vehicles and it is of particular importance to obtain the paths of the vehicles' motions. To match the objects in the current frame, using the detected objects in the previous frames, one may use one of the two methods of the texture feature matching and the Bayesian method. We can use the algorithm of texture feature matching between the image sequences to obtain the moving objects' paths.

Similar to Equation (3) (shadow detection), for the shadow-removed moving region (SRMR) n in the image L, we can match the texture feature with all SRMR in the image $L-1$. If there is a SRMR m, whose texture feature is matched with those of region n and $Sim < TH$, we can conclude that the vehicle in SRMR n is the same vehicle in SRMR m. [32].

To analyse the observed images and to match the detected objects in the current image, given the objects in the previous image, we utilize the classical Bayesian approach, as one of the contributions of this paper is due to its significance on how it is combined with other components to make the best use of a priori information to obtain a more robust detection of salient regions.

In particular, to model the uncertainty in associating an observed image to a detected object of the previous image (indexed by α), we derive a posterior density subject to this unknown. The posterior density is related to the product of likelihood and prior probability [34, p. 874]:

$$P(I^\alpha|I^D) = \frac{1}{P(I^D)}P(I^D|I^\alpha)P(I^\alpha), \qquad I^\alpha \in \mathcal{A} \tag{6}$$

where I^D is the detected object in the current image and $\mathcal{A}$ is the set of all the detected objects in the previous image.

$P(I^\alpha)$ incorporates our prior knowledge on finding a target I^α, which has a certain pose and a location in the scene. For example, in the case of moving targets, the knowledge of target location may imply a higher probability of the existence of a future target presence in certain areas and a lower probability in the rest [34, p. 874]. One function that can be used for modeling is a 2-D Gaussian distribution around the center of each detected object in the previous image frame. The likelihood function $P(I^D|I^\alpha)$ quantifies the probability that a target I^α will give rise to the observed image I^D. Assuming the difference between the two has Gaussian distribution,

$$P(I^D|I^\alpha) = \frac{1}{(2\pi\sigma^2)^{d/2}} exp(\frac{-1}{2\sigma^2}\|I^D - I^\alpha\|). \tag{7}$$

The resulting posterior includes all the information we need for target recognition. We now focus on the main task: finding the index α that best matches a given image I^D. As described earlier, in a Bayesian framework, the estimated target type is given by the index with maximum a-posteriori probability. It becomes an M-array hypothesis test that is:

$$\hat{I^\alpha} = \underset{\alpha \in \mathcal{A}}{argmax} P(I^\alpha|I^D) \tag{8}$$

where the posterior is calculated by using the Bayes' rule:

$$P(I^\alpha|I^D) = \frac{P(I^D|I^\alpha)P(I^\alpha)}{P(I^D)} \tag{9}$$

4) Likelihood-based Algorithm: There are several methods previously used to track and find objects in sequential frames such as:

 (i) BMMC (block-matching motion compensation): to find the object in the new frame.
 (ii) Condensation alignment (particle filtering): to track over long video sequences.
(iii) Active appearance model: to detect the corresponding points in consecutive frames (even after resizing and rotation)

All these methods are search-based and therefore computationally very intensive. Hence, similar to [27, 28], we propose a new algorithm based on Bayesian method which alleviates this problem.

Figure 5 shows a sample of the region bounds where

$$i_b = 1, ..., N_S, \ j_c = 1, ..., M_S \ \ \forall b, c \tag{10}$$
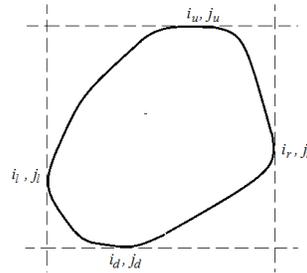
Figure 5: A shapeless region and its left-right and top-bottom bounds.

First, the center points of detected objects in previous frame are calculated:

$$i_{old} = \frac{i_u + i_d}{2} \tag{11}$$

$$j_{old} = \frac{j_l + j_r}{2} \tag{12}$$

Next, the priori probability is computed. In order to compute this probability, Gaussian kernel is used. The center points of detected objects in current frame are calculated in the same fashion as above.

The posteriori probabilities can be calculated by using the distance between the center points of the objects in previous and current frames and using Gaussian kernel. Based on posteriori matrix, in each column, the maximum value and its index are calculated and we find relations between objects in the previous and the current frames.

Here unlike [16] in the recognition step, we use Bayesian method which ultimately results in speeding up the algorithm.

C. Localization
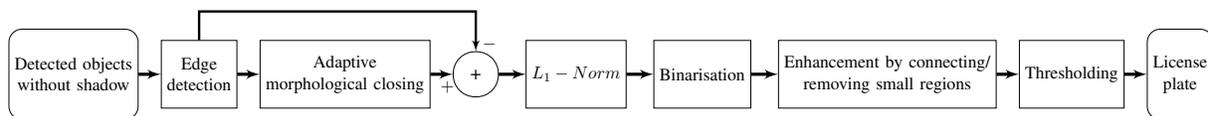
Figure 6: The block diagram of the localization process.

After we detect the objects and remove the shadows, we find the coordinations of LP in each object, detected in the tracking step, as the goal of localization step. The block diagram of the localization operation is shown in Figure 6. As shown in this figure, the first step is to detect the vehicles after the tracking step discussed previously. Then, the Sobel method is applied to detect the edges of the ROI and

(a)   (b)
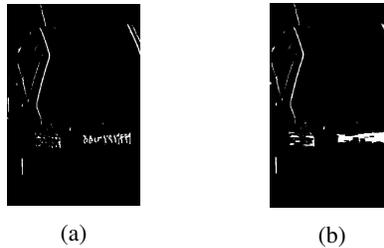
Figure 7: The result of (a) closing using fixed size and (b) closing using adaptive size.

the edge segments are connected using a rectangular structure element of size $[S_P1, S_P2]$. S_P1 and S_P2 are set adaptively based on the input image size (i.e, for our dataset 1),

$$
\begin{aligned}
S_P1 &= 2 \\
S_P2 &= min(20, max(10, [CulNum/50]))
\end{aligned}
\tag{13}
$$

where $CulNum$ denotes the number of columns of input image.

Figure 7 compares the closed edge image using fixed and adaptive element sizes. By differencing edge image from the closed edge image, we remove the solitary edges. Then, $L_1 - Norm$ operation calculates sum of the neighborhood pixels:

$$
\|IDiff(m,n)\|_1 = \sum_{i=m-M_W}^{m+M_W} \sum_{j=n-N_W}^{n+N_W} |IDiff(i,j)|
\tag{14}
$$

where M_W and N_W determine the size of the window around the current pixel and $IDiff$ denotes the difference image calculated in the previous step. In the next step, the $L_1 - Norm$ image is binarized. The Otsu's method is used to set the threshold level. Using the histogram, the image contrast is also enhanced. The LP candidates satisfy two conditions: First, the ratio has to be bigger than the minimum LP aspect ratio. Second, it has to have at least six peaks and six valleys in its vectorized edge map. Based on the edge image of these candidates, their regions may be resized. Comparing to [30, 31] and the research carried out by Kanani et al. [35] that uses wavelet decomposition for localizing the LP or even the earlier work of our own [9], which uses fixed closing size for the edge image, our proposed localization method uses adaptive size in closing, which improves the accuracy.
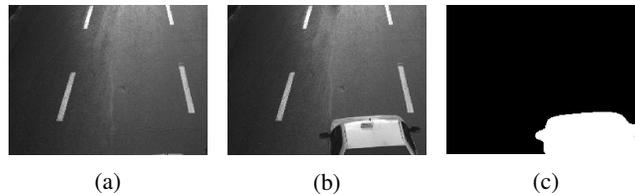
Figure 8: 3D-saliency in video frames; (a) background image, (b) current image, and (c) the salient object is shown by white region.

III. EXPERIMENTAL RESULTS

A. Scope of evaluation

To evaluate our proposed tracking and localization algorithm, we tested the method over three datasets. Dataset 1 has 200 images with an image size of 1280×960 pixels. The images of this dataset were taken by infrared (IR) camera. The background images in dataset 1 were categorized as "simple". The second dataset contains colour images with 300 frames with size of 1294×964 pixels. Since the background changes in time, it may be classified as "complex". Dataset 3 has 200 frames, each of size 1280×960 pixels. IR camera was used to capture these images and it may be classified as "IR images". The images in this dataset were brighter than those of the first dataset. Our proposed algorithm was tested on Core-i7 CPUs with clock rate of 2.30 GHz. For the evaluation of the algorithm, we considered accuracy (ACCY), computation time (CT) and false alarm rate (FAR) for object tracking, shadow removal, object recognition and localization.

B. Results

Figure 8 shows the outcome of the 3-D saliency-based method when applied to video frames and Figure 9 shows some results of the background extraction experiment.

Figure 10a shows binarisation of the difference image and Figure 10b shows binarisation of the $L_1 - Norm$ image. The binary image is then vectorized. The vectors with high numbers of foreground pixels were then determined as belonging to the ROI. As shown in Figure 10c, in the last step of the object detection algorithm using CCA, each object can be detected.

Table II: Experimental results for the proposed algorithm using three datasets.

Datasets	Object Detection			Shadow Detection			Object Recognition			Localisation			Total		
(Number of frames)	ACCY	CT(ms)	FAR	ACCY	CT(ms)	FAR	ACCY	CT(ms)	FAR	ACCY	CT(ms)	FAR	ACCY	CT(ms)	FAR
1(200)	100%	13.8	10.6%	76.8%	12.1	4%	94.9%	0.1	4.2%	100%	34.3	6.1%	92.92%	60.3	25.6%
2(300)	100%	12.7	10.8%	78.6%	11.5	5.1%	91.6%	0.1	5.5%	98.3%	36.5	7.4%	92.1%	60.8	26.7%
3(200)	99.7%	14.4	18%	74.8%	12.5	4.7%	93.7%	0.1	6%	99.1%	32.1	18.4%	91.8%	59.1	23.6%

(a)   (b)   (c)
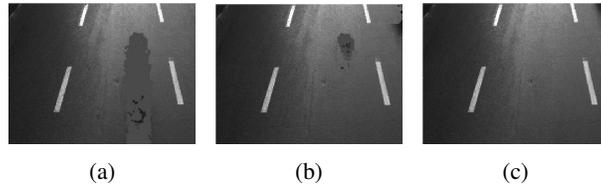
Figure 9: Experimental results of the background extraction. Background extracted at the $1^{st}(a), 8^{th}(b), 13^{th}(c)$ frames of the video sequence.

(a)   (b)   (c)
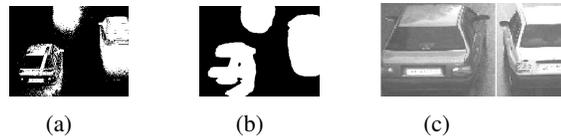
Figure 10: (a) binarized difference image, (b) binarized norm image, and (c) detected objects.

Figure 11 shows the results of shadow detection using texture analysis method. As shown in this figure, the method can not detect the shadows and determines just some parts of the black objects as shadows.

Figure 12a shows the detected object with shadow and Figure 12b shows the detected white object without the shadow when our proposed shadow removal method was applied.

The outputs of object detection for the first dataset is shown in Figure 13 (a). All moving vehicles were detected, although in few cases, the objects were merged and selected as one object. This is mainly because there were shadows between the vehicles. Figure 13 (b) shows the results of shadow detection

(a)   (b)
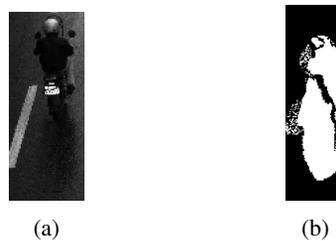
Figure 11: Result of shadow detection using texture analysis.

(a)   (b)
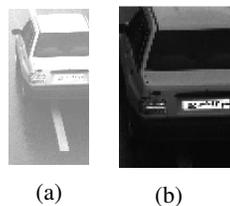
Figure 12: Detected object without shadow.

(a) Output of object detection   (c) Output of object recognition

(b) Output of shadow detection

(d) Output of localisation
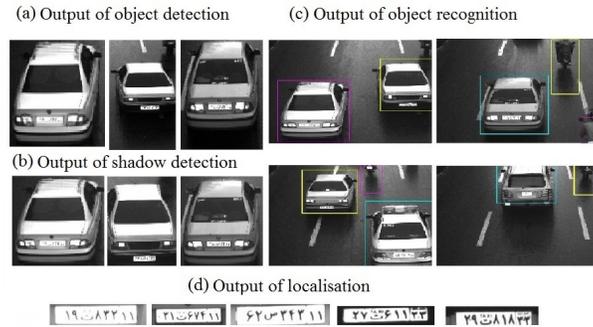
Figure 13: Results of applying the proposed algorithm.

Approach	Accuracy	Computation time (ms)
Wang et al. 's [17]	92%	176.7
Jiang et al. 's [19]	82%	250
Proposed algorithm	100%	13.8

Table III: Comparison of our proposed object detection method to other video-based methods using dataset 1.

and removal within the object detection step of the first dataset. As shown in this figure, the shadows of white objects are eliminated perfectly and the objects connected by shadows are detected separately. As shown in Table II, the FAR in this step is severely reduced from 10.6% to 4%. This is mainly because of the eliminated objects similar to the background block as the result of shadow detection. However, the shadows of black objects can not be detected. There are two reasons for the decrease in the accuracy of shadow detection for dark objects:

- Shadow of black object is difficult to detect.
- When objects are going to exit the coverage area, they can not be detected as white objects. So, the proposed algorithm detects them together with their shadows as black objects.

The outputs of the object recognition for selected images of the first dataset are shown in Figure 13 (c). Since in this step, there is no detection, FAR can not be measured. The accuracy of this step is 94.9%, that is considered to be a high recognition rate. Error occurs when an object exits the coverage area and another object is about to enter into that same area. In this state, the proposed algorithm cannot decide whether a new object has entered, neither it can set the same label to the new object as the old one. The LPs in the selected frames of dataset 1 are shown in Figure 13(d). All LPs of this dataset are detected with 100% accuracy. However, there were some false detections in this step. It should be noted that these wrongly-detected frames are very similar to LPs and most of them were removed in the post processing steps.

(a)   **(b)**   **(c)**   **(d)**
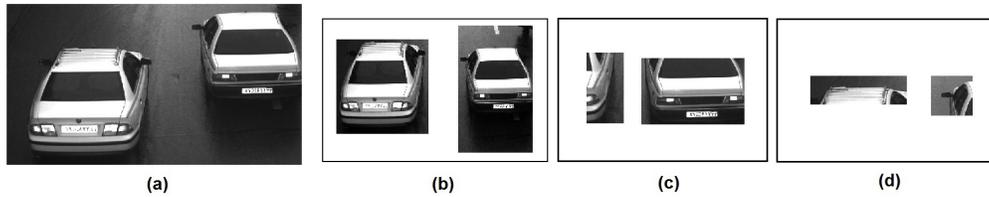
Figure 14: (a) sample image, (b) result of our proposed object detection algortim, (c) result of [17]'s object detection algortim, (d) result of [19]'s object detection algortim.

Approach	Accuracy	Computation time (ms)
Russell et al. 's [20]	74%	52.5
Zhu et al. 's [33]	71%	60.5
Proposed algorithm	76.8%	12.1

Table IV: Comparison of our proposed shadow detection method with other methods using dataset 1.

(a)   **(b)**   **(c)**   **(d)**
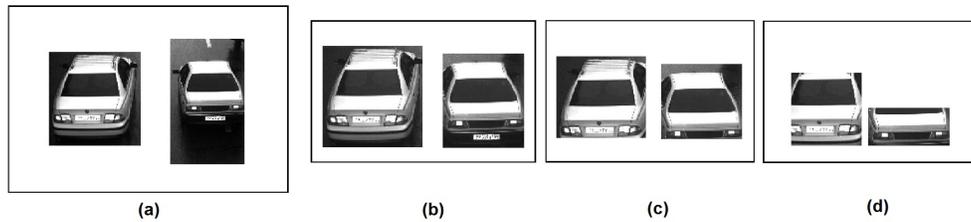
Figure 15: (a) sample image, (b) result of our proposed shadow detection algortim, (c) result of [20]'s shadow detection algortim, (d) result of [33]'s shadow detection algortim.

Approach	Accuracy	Computation time (ms)
Kanani et al. 's [35]	92%	114
Safaei et al. 's [9]	85%	120.5
Proposed algorithm	100%	34.3

Table V: Comparison of our proposed localization method with other methods using dataset I.

(a)   **(b)**   **(c)**   **(d)**
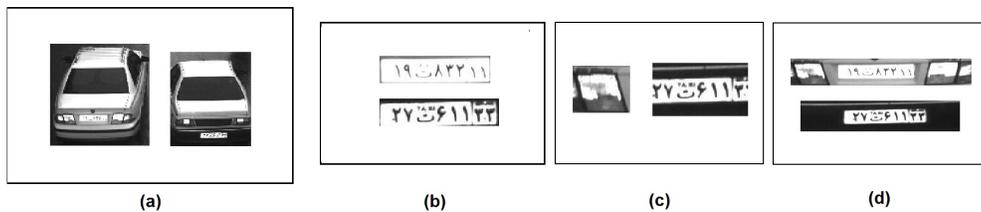
Figure 16: (a) sample image, (b) result of our proposed localization method, (c) result of [35]'s localization method, (d) result of [9]'s localization method.

(a)   (b)
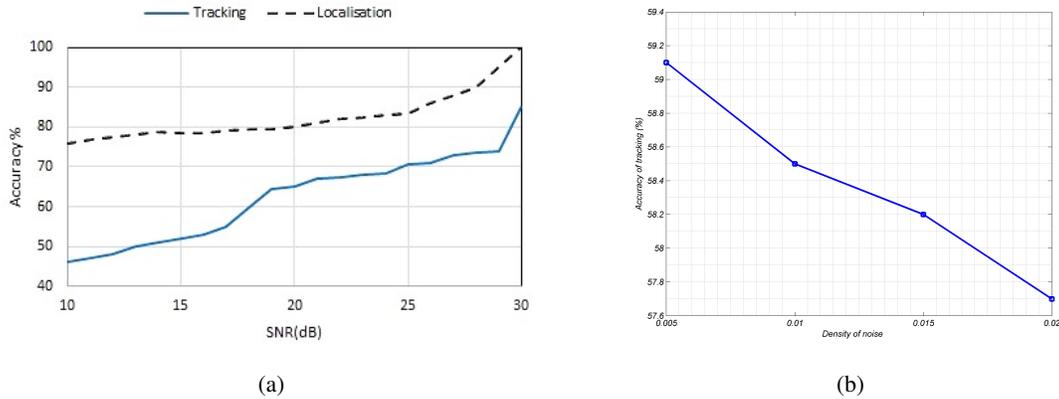
Figure 17: Accuracy of the tracking and localization steps.

As shown in Figure 14, the algorithms developed by Wang et al. [17] and Liu et al. [19] have a high false alarm rate and low accuracy in object detection. Figure 15 shows the outcome of our proposed method comparing to those developed by Russel et al. [20] and Zhu et al. [33]. As shown in this figure, in [20] some parts of objects are eliminated when the shadow detection and removal algorithm is applied. Moreover, the shadow detection proposed in [33], with more false shadow detection, has removed a larger part of the object instead of the shadows. Figure 16 presents the results of applying our proposed algorithm comparing to other methods. As shown in this figure, the method used by Kanani et al. [35] has detected some wrong LPs. Although in [9] the LPs are detected correctly, the localization algorithm does not detect the right positions of the merged (tight) number plates. Tables III, IV and V show the results of evaluation and comparison of our proposed method with those of other recent methods using dataset I. As shown in these tables, the superiority of our proposed methods in terms of accuracy and computation time is notable.

C. Evaluation

To evaluate the robustness of our proposed method, two types of noise were added to the images: Gaussian and salt & pepper. Figure 17a shows the accuracy of tracking versus signal-to-noise ratio (SNR) dB for Gaussian noise. When the Gaussian noise power is increased (SNR is decreased), the accuracy of tracking step is reduced. The Accuracy of localization step for different values of SNR is shown in Figure 17a. Note that the accuracy of localization is higher than 75% even for SNR $= 10$ dB. However when SNR is lower than 20 dB the accuracy of tracking step reduces significantly. This can be improved by low-pass filtering the images before hand and using an adaptive threshold.

Parameter	background Extraction					Shadow Detection										Localisation				
	TH value					TH value					T1,T2 vlaue					S1,S2				
Parameter	0.01	0.02	0.03	0.04	0.05	0.02	0.04	0.06	0.08	0.1	[0.01,1.05]	[0.01,1.02]	[0.05,1.05]	[0.01,1.08]	[0.05,1.08]	[2,-]	[3,-]	[4,-]	[5,-]	[6,-]
ACCY	100	100	96	96	94	39	44	45	48	48	78	61	62	62	60	100	100	98	98	95
CT(ms)	16	13	13	13	12	115	120	124	132	150	12	12	12	12	12	34	35	35	35	36

Table VI: Effect of parameters on ACCY and CT.
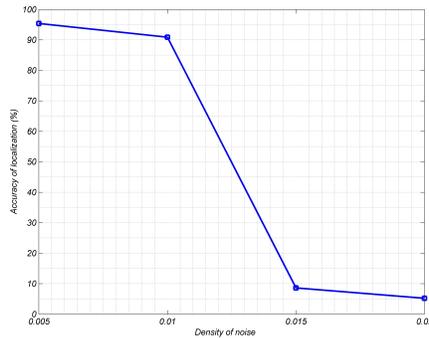
Figure 18: Accuracy of the localization step versus noise density when salt & pepper noise is added.

Rain or snow can significantly affect the accuracy of ALPR systems. In order to measure the affect of these types of noises, salt & pepper noise can be used. The accuracy of the tracking step versus the noise density is shown in Figure 17b. When the noise density is increased, the accuracy of the tracking step is reduced slightly. The accuracy of the localization step for different values of noise densities is shown in Figure 18. Note that the accuracy of the localization is reduced significantly by increasing the noise density.

Evaluating the affect of choosing different thresholds and parameters (i.e. background extraction, shadow detection and localization) in CT and ACCY was carried out and is shown in Table VI. As shown in this figure, when TH value in background extraction is increased ACCY and CT are reduced. In shadow detection step, TH value has also affect on CT and ACCY. However changing $T1$ and $T2$ value of shadow detection is not affecting CT, although ACCY is reduced. On the other hand in localization step increasing $S1$ value slightly affects CT and ACCY (note that S2 is adaptive with number of columns of input image).

IV. CONCLUSIONS

In this paper, we proposed new algorithms for tracking and localization of LPs. The tracking involves three fundamental parts: object detection, shadow detection, and object recognition. For object detection, the background image was calculated and the moving objects were detected. To remove the shadows, we

proposed a new shadow detection algorithm to discriminate shadows from their corresponding objects. In the last step of tracking, the relation between the objects in sequential frames was determined using a Bayesian framework. Each of the detected objects was processed after the shadow removal, to find the LPs. The results of applying the method to three different datasets were presented. Object detection, object recognition and localization exhibit a high accuracy and the shadow detection algorithm exhibits an accuracy of about 70%. In order to evaluate the robustness of the proposed system, we added Gaussian and salt & pepper noises to the input images. The accuracy reduces when the noise level increases. However, the localization accuracy was higher than 91%. When salt & pepper noisy images were processed, the localization accuracy reduced significantly. However, the tracking approach exhibited acceptable accuracy in different noise power scenarios. Also, unlike in our previous method [9], in the proposed method in this paper, the backgrounds of the images were more accurately extracted. Moreover, the object shadows were removed by an accuracy of approximately 70%. Object recognition step was added to the algorithm which introduced a good performance. In the localization step, the 3-D salient regions were detected, instead of 2-D ones. In addition, an adaptive closing algorithm was used in the localization step.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(2):311–325, 2013.

[2] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas. A license plate-recognition algorithm for intelligent transportation system applications. *Intelligent Transportation Systems, IEEE Transactions on*, 7(3):377–392, 2006.

[3] J. Matas and K. Zimmermann. Unconstrained licence plate and text localization and recognition. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 225–230, Sept 2005.

[4] A. Broumandnia and M. Fathy. Application of pattern recognition for farsi license plate recognition. *ICGST International Journal on Graphics, Vision and Image Processing*, 5(2):25–31, 2005.

[5] R. Zunino and S. Rovetta. Vector quantization for license-plate location and image coding. *IEEE Transactions on Industrial Electronics*, 47(1):159–167, Feb 2000.

[6] H. J Lee, S. Y. Chen, and S. Z. Wang. Extraction and recognition of license plates of motorcycles and vehicles on highways. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 356–359. IEEE, 2004.

[7] I. K. Yalcin and M. Gokmen. Integrating differential evolution and condensation algorithms for

license plate tracking. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 658–661, 2006.

[8] M. Isard and A. Blake. Condensationconditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[9] A. Safaei, HL. Tang, and S. Sanei. Search-free license plate localization based on saliency and local variance estimation. In *Seventh International Conference on Machine Vision (ICMV 2014)*, pages 94451U–94451U. International Society for Optics and Photonics, 2015.

[10] A. Safaei, HL. Tang, and S. Sanei. Robust search-free car number plate localization incorporating hierarchical saliency. *J Comput Sci Syst Biol*, 9:093–103, 2016.

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[12] S. Shantaiya, K. Verma, and K. Mehta. Multiple object tracking using Kalman filter and optical flow. *European Journal of Advances in Engineering and Technology*, 2(2):34–39, 2015.

[13] T. Kodama, T. Yamaguchi, and H. Harada. A method of object tracking based on particle filter and optical flow. In *ICCAS-SICE, 2009*, pages 2685–2690. IEEE, 2009.

[14] A. Li and S. Yan. Object tracking with only background cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1911–1919, 2014.

[15] S. Aslani and H. Mahdavi-Nasab. Optical flow based moving object detection and tracking for surveillance. *International Journal of Electrical, Electronics, Communication, Energy Science and Engineering*, 7(9):789–793, 2013.

[16] P. K. Bhaskar and S. Yong. Image processing based vehicle detection and tracking method. In *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, pages 1–5. IEEE, 2014.

[17] H. Y. Wang, L. H. Wang, and C. B. Wu. An efficient background extraction and object segmentation algorithm for realtime applications. In *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on*, pages 659–662. IEEE, 2012.

[18] C. Chiu, M. Y. Ku, and L. W. Liang. A robust object segmentation system using a probability-based background extraction algorithm. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(4):518–528, 2010.

[19] S. Jiang and Y. Zhao. Background extraction algorithm base on partition weighed histogram. In *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, pages 433–437, 2012.

[20] M. Russell, J. J. Zou, and G. Fang. Real-time vehicle shadow detection. *Electronics Letters*,

51(16):1253–1255, 2015.

[21] X. Yuan, M. Ebner, and Z. Wang. Single-image shadow detection and removal using local colour constancy computation. *IET Image Processing*, 9(2):118–126, 2014.

[22] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li. Video processing techniques for traffic flow monitoring: A survey. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1103–1108, 2011.

[23] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

[24] G. Xiang. Robust face tracking based on Bayesian probability. In *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, pages 3–985–3–988, Aug 2009.

[25] K. G. Jamieson, M. R. Gupta, and D. W. Krout. Sequential Bayesian estimation of the probability of detection for tracking. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 641–648, July 2009.

[26] L. Stone, R. Streit, T. Corwin, K. Bell, and F. Daum. Bayesian multiple target tracking, 2nd edition [book review]. *IEEE Aerospace and Electronic Systems Magazine*, 29(8):23–24, Aug 2014.

[27] S. Zhang, S.C. Chan, B. Liao, and K. M. Tsui. A new visual object tracking algorithm using Bayesian Kalman filter. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 522–525, June 2014.

[28] P. Mazurek and K. Okarma. Vehicle tracking using a multi-scale Bayesian algorithm for a perspective image of a road. In *International Conference on Transport Systems Telematics*, pages 346–353. Springer, 2011.

[29] P. I. Reji and V. S. Dharun. License plate localization: A review. *International Journal of Engineering Trends and Technology (IJETT)*, 10(13):604–615, 2014.

[30] S. M. Roomi, M. Anitha, and R. Bhargavi. Accurate license plate localization. In *IEEE Computer, communication and Electrical Technology (ICCCET), 2011 International Conference on*, pages 92–97, 2011.

[31] D. Wazalwar, E. Oruklu, and J. Saniie. Design flow for robust license plate localization. In *Electro/Information Technology (EIT), 2011 IEEE International Conference on*, pages 1–5, 2011.

[32] D. Luo and X. Huang. Texture analysis for shadow removing and tracking of vehicle in traffic monitoring system. In *International Symposium on Intelligent Information Technology Application Workshops*, pages 863–866. IEEE, 2008.

[33] Z. Zhu and X. Lu. An accurate shadow removal method for vehicle tracking. In *Artificial Intelligence and Computational Intelligence (AICI), 2010 IEEE International Conference on*, volume 2, pages

59–62, 2010.

[34] A. C. Bovik. *Handbook of image and video processing*. Academic press, 2010.

[35] P. Kanani, A. Gupta, D. Yadav, R. Bodade, and R. B. Pachori. Vehicle license plate localization using wavelets. In *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, pages 1160–1164, 2013.

Amin Safaei received his B.Sc. from Northumbria university, UK and M.Sc. in software engineering from Sunderland University, UK. He is currently working toward his Ph.D. in the Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, U.K. His main research interests include image and video analysis, object tracking, number plate recognition, software engineering and data warehousing.

H. Lilian Tang received her PhD in medical informatics from the University of Cambridge, United Kingdom. She is currently a senior lecturer in Faculty of Engineering and Physical Sciences, University of Surrey, UK. Her research interests include image analysis and vision, object tracking, data analytics, statistical methods and learning, mostly with applications to medicine, surgery and microbiology.

Saeid Sanei received his PhD in signal processing from Imperial College London, UK. He has published three monograms and over 330 papers in peer reviewed journals and conference proceedings. He has served as an Associate Editor for the IEEE Signal Processing Letters, IEEE Signal Processing Magazine, and Journal of Computational Intelligence and Neuroscience. His research encompasses adaptive and nonlinear multichannel, multimodal, and multi-way signal and image processing.