On-road Night-time Vehicle Light Detection and Tracking Methods Overview

Darko Jurić

Abstract—One of the main issues during night-time driving is a good visibility of the road ahead. Although the traffic volume is much lower during night-time, the percentage of traffic accidents is much higher. High-beams are used too rarely or in non-appropriate situations thus dazzling other drivers, which can be lethal. In urban areas there is an increased need for attention which can lead to an accident when mixed with driver fatigue and unpredictable preceding vehicle movements. Therefore advanced driver assistance systems (ADAS) can lead to more safety driving by automatically controlling headlights or estimating a distance between vehicles. Headlight control system demands an early vehicle detection where only camera based approaches are suitable.

In this paper the recent methods that rely on single camera input are described. The paper is divided into several sections that describe camera hardware requirements and used methods for vehicle light detection, classification and tracking. In the end current state-of-the-art results are presented along with problems that are remained to be solved.

Keywords—nighttime vehicle detection, light blob detection, image segmentation, object tracking, object classification

I. INTRODUCTION

Despite the fact that the traffic volume is much lower, 42 percent of all traffic accidents occur after dark, 58 percent are fatal accidents and 67 percent are pedestrian fatalities [1].

Advanced driver assistance systems (ADAS) could help to reduce those numbers. Those systems use cameras to gain information from visible spectrum, radars to estimate distance between vehicles or both. The cameras have an advantage over radar based systems because vision based hardware have greater field of view [2] and enable us to spot vehicles at much greater distances. Headlight and tail-light detection systems are an important part of advanced driver assistance systems. The most common systems built on top of vehicle light detection and tracking are headlight control and forward collision warning system (see Figure 1).

A. Headlight control

Vehicles are equipped with variety of different lamps to provide proper illumination for the driver and serve as visible markings for other drivers. Headlights usually have two modes: high-beam and low-beam lights which are much more frequently used. See Figure 1.

Fig. 1: Two ways of headlight beam control [3]. Top: Low and high beam. Middle: Adaptive cut-off beam Bottom: Fully adaptive beam control (e.g. by using LEDs)
(a) Far vehicle detection for headlight control [4].

(b) Rear-light detection and tracking for time to collision estimation [5].

(c) Vehicle light detection for distance and time to collision estimation [2].

Fig. 2: Night-time vehicle detection systems.

and MobilEye [10]. SmartBeam™ consists of a specialized sensor which is mounted behind the rear-mirror. It detects headlamps for oncoming vehicles and tail-lights for preceding traffic. Its purpose is headlight mode switching. MobilEye is also a system for adaptive headlight control. The operation distance for headlight detection is up to 800 m and up to 400 m for tail-lights.

B. Forward collision warning

Driver distractions and unpredictable preceding vehicle movements especially during night can lead to accidents. The number of rear-end collisions is about 15 percent in Germany [11] and about 25 percent in the USA [12]. Although rear-end collisions have less casualties than frontal collisions, other non-direct damage can occur such as causing traffic jams and thus wasting other resources.

Although some vehicles are equipped with adaptive breaking lamps which blink if the negative acceleration is too big [2], rear-end accidents can still happen. Forward collision warning systems can prevent such accidents. Such system use radars, cameras or both. Because this kind of systems are usually designed for urban areas, vision-based systems have issues due to highly lit environment, where the lighting sources comprise of large amount of street-lights, building lights.

II. PAPER OVERVIEW

The paper serves as an overview of recent methods for night-time vehicle detection using camera sensors. The generic pipeline which those methods usually follow as the standard pattern for headlight/tail-light night-time detection is shown in Figure 3.

Fig. 3: Common pipeline of methods for light detection and tracking

The pipeline starts with image preprocessing (e.g. Gaussian blur, red channel extraction, defining region of interest). Next, lights candidates are detected. After light candidates are detected classification can use light pairing as a standalone feature which in this case is mandatory, or as one of features to achieve better performance. Pairing is a strong feature that can reduce false positives and can help to estimate vehicle distance. Tracking procedure is usually heavily interleaved with detection and classification. Tracking can be based on set of rules that are specific for vehicle light tracking. Those steps form the base for other systems. The additional system parts rely on the base to access all already preprocessed information, so those parts of a system are labelled as system dependent logic.

The paper is organized as follows: In section III considerations which should be taken during camera sensor selection are described. Section IV describes methods for light blob candidate detection based on monocular camera. Section V shows two approaches for vehicle light selection from detected light blob candidates. Tracking methods are described in Section VI. In Section VII representative method results are mentioned. Some custom performance measurements are also briefly described. Finally, Section VIII gives the summary and the conclusion.

III. CAMERA PARAMETERS

There are several important camera parameters which should be taken into consideration: sensor type, resolution of a camera and lens.
A. Sensor type

There are several types of camera sensors that are used for night–time vehicle detection:

1) Monochrome: Monochrome sensor has the best sensitivity but it is unable to extract the color information. It is used in \cite{4} and \cite{13}.

2) Color (BGR): The most common sensor is the three channel color sensor \cite{14, 5, 15, 16, 17, 18}. More recent papers use HDR color sensors like in: \cite{2, 19} which uses HDR color sensor with 10 bits per channel, \cite{20} which fixes exposition rate to smear distant spots in order to have better color rendering and to smooth out AC and LED light pulsating, and \cite{21} where manual exposition and white balance are set.

3) Read-monochrome Bayer: Red-monochrome Bayer sensor consists of 75 percent monochrome pixels and 25 percent red pixels. That way it can compensate poor sensitivity of RGB and Bayer sensors and can help to separate between tail-lights and headlights by incorporating red color information. This sensor is used in \cite{22}.

4) Specific: SmartBeam by Gentex \cite{23} which is developed to smooth out AC and LED light pulsating, and \cite{21} which fixes exposition rate to smear distant spots in order to have better color rendering and to smooth out AC and LED light pulsating, and \cite{21} where manual exposition and white balance are set.

B. Resolution

Used resolutions are: 360 x 240 in \cite{18}, 752 x 480 in \cite{22, 5, 15, 16, 17, 20, 2, 4, 13}, 720 x 576 in \cite{14, 21} and 640 x 480 in \cite{19}. Width of the camera sensor is larger than normal in order to accommodate twisting roads \cite{20}.

C. Lens

A study in \cite{4} regarding oncoming and proceeding vehicle detection has been made. As a result optimal lenses focal lengths are obtained: 4.3 mm – 6 mm. Lens with 4.3 mm focal length is chosen because the lens field of view is close to the field of view of human

D. Camera calibration

Singe camera system calibration consists of fixing three-dimensional position in real-world. Fixing those parameters and by assuming distance between vehicle lights or vehicle vertical ground position is the first step of calculating distance between vehicles. Most papers using perspective camera models assume a road model with 0 tilt angle. Papers \cite{13} and \cite{4} extends the perspective projection model \cite{24} to approximate road vertical and horizontal curvature. The performance of classification that relies on object position can be improved by correcting these two parameters \cite{4}.

IV. VEHICLE LIGHT DETECTION

There are several methods for light blob detection, which are going to be mentioned in this section, that are commonly exploited in this problem. Regarding embedded device performance constraints those methods must be light-weight in performance terms, but in the same time robust enough to achieve the desired task of extracting vehicle light candidates. The most common methods are based on some kind of thresholding.

A. Simple thresholding

In \cite{19} and \cite{22} the simple fixed thresholding is used. To detect components a connected component analysis is performed.

After applying thresholding some lights stay joined together. In \cite{14} this case is resolved in a way that threshold is risen until blobs are split or the maximum value is reached. The blob splitting is done during the tracking phase. Other post-processing techniques include non-maxima suppression \cite{25}.

B. Adaptive thresholding

In \cite{13} adaptive thresholding is applied to an input image in order to detect headlights. Authors show a typical headlight structure and say that this structure has Gaussian shape where center pixel values are above 250, and edge pixels have intensity lower than 50. Other structures do not have such high-valued center pixels, except some on-road reflections.

In \cite{4} light blobs are detected in the following way: A low threshold value is used to extract potential objects. Then mean \cite{2} and standard deviation \cite{3} of the grey level of an image is computed. An adaptive threshold value \cite{1} is applied.

\[
\text{Threshold} = \mu - k \cdot \sigma
\]  
\[
\mu = \frac{1}{N} \sum_{i=1}^{N} \mu_i
\]  
\[
\sigma = \frac{1}{N} \sum_{i=1}^{N} \sigma_i
\]

Parameter \(k\) is fixed, but can be used for tuning. The paper also suggests scale normalized Laplacian to extract blob like objects. To extract headlights and tail-lights two scales are used. This procedure however is not chosen due to real-time restrictions. To reduce computation complexity two horizontal cut-off lines are used in the image.

A method which performs multi-thresholding is presented in \cite{20}. A multi-thresholding technique with region-growing has been used to detect spot lights. Thresholding is done in red channel of the image in 8 levels. Paper does not recommend spreading thresholds uniformly over the whole interval, but rather to put more thresholds (group them) at the bottom end (far tail-light separation) and at the top end (near headlight separation). Authors claim that the connected component computations with 8 threshold does not take considerably longer compared to processing with a single threshold if the method is implemented carefully. A region growing is performed for each blob until a condition for minimum blob size is not satisfied or a growing region is not merged with another one.
1) Hardware support: The paper [26] proposes a novel system that combines different camera exposures and algorithm that successfully uses multi-image information. The first image is obtained using an automatic exposure (AE) setting, and the other one by using low exposure (LE) on the same camera.

AE is used to detect far headlights and distant tail-lights, and the LE image serves for detecting distant high-beams and near headlights along with near tail-lights. Lane detection, which is obtained by using AE image, serves to position far region of interest - FROI where distant vehicles are likely to exist. Light blobs in FROI are extracted using Laplacian of Gaussian - LoG at one scale followed by thresholding. For LE images adaptive thresholding defined in (1) is used. A direct relationship between blob’s vertical image coordinate and intensity which depends on camera exposure is also used.

C. LoG approximation

In [18] a technique based on Laplacian of Gaussian - LoG approximation is presented - CenSurE - Center Surround Extremas. Before the blob detection method is applied, the image is first split to RG where R and G represent red and green channel respectively and the red image which is obtained by the equation (4).

\[
Red = (R - G) \cdot \alpha + \beta
\]  

Parameters R and G correspond to image channels. Constants \(\alpha\) and \(\beta\) are fixed to 1.

On each image the blob extraction algorithm is applied. An integral image is made and the approximation of the LoG operator is applied on several scales.

In [16] a RGB-to-Y (from Yuv) color space is used to convert an input image to intensity image. Then an adaptive thresholding is performed.

Paper [21] detect only tail-lights by HSV thresholding. HSV thresholds are derived from automotive regulations. To adopt to real-world images channel thresholds are computed from image training data.

D. Particle filtering

In [27] as preprocessing step the gradient image of an input image is calculated. Features are extracted from an image and their values are combined into single weighting function for each particle. Blob detection is done by using multiple particle filters.

V. BLOB PAIRING AND CLASSIFICATION

After light candidate detection process, blobs are classified as vehicle and non-vehicle lights. To reduce false positives light pairing is used very often as a standalone feature or in a classification process, which uses more features where light pairing is not a mandatory. Papers that rely on light pairing can detect only vehicles that have more than one light, thus the detection process does not apply to motorcycles. The section divides current work to methods that do require light pairing and the ones for which that feature is not mandatory.

The section starts with feature selection that is commonly used. Most of them are features that are obtained by calculating characteristics of binarized blobs.

A. Features

1) Image extracted features: Paper [13] exploits properties of halo-effect where black hat image transformation is calculated as shown in Figure 4. Hat value is obtained as an average pixel intensity of the detected object in black-hat transformed image. This image exploits properties of halo-effect, where the operation is defined as residue between an input image and a closing information where local valleys are obtained as shown in equation (5).

\[
BH_T(x, y) = (f - f \circ B)
\]

where \(f\) is a gray scale image and \(B\) is a structuring element.

Other features that utilize pixel intensities in direct way are [20]: averaged H, S, V channels, highest value in blob, H, S, V for halo effect, halo area and area without halo ratio.

2) Binarized blob features: Features that are used for classification are most commonly extracted from already binarized blobs. In [13] and [4] they include: area in pixels, centroid coordinate, rectangularity, aspect ratio, contour length, circularity. The Hue moments are also used. Road vertical and horizontal correction shortens the time necessary to classify the vehicle lights correctly.

Binary image features are also used in [20] along with some features that are extracted directly from image pixels: area, compactness \(\text{perimeter}^2/\text{area}\), bounding box center position, elongation and fill ratio.

3) Motion features: In [2] among binary image based features similar to already mentioned ones, motion features are also used. Vertical as opposed to horizontal movement, which will differ due to distance change and vehicle position, will always stay similar. The deviation between predicted and
actual spot position and the vertical similarity are used to reduce false positives. All factors are multiplied and compared with predefined threshold. In [19], a total of five motions features are extracted: horizontal and vertical speed magnitude, direction and motion displacement.

**B. Pairing**

In [2] and [28] clustering and classification process has two steps: pre-selection and pairing. Pre-selection is done by using constraints such as that blobs must be on the same height, the height of bounding region that contains vehicle light blobs must be at least 2 times smaller than its width. Pre-classification step excludes blobs that disappeared for few frames and pairs that do not satisfy the mentioned constraints.

Classification process uses the following similarities: covariance intersection - fitting ellipses on blobs and calculating ellipse area intersection. Motion features are also used like in [V-A]. All factors are multiplied and compared with predefined threshold. A camera roll pitch factor is also included to correct possible camera roll and pitch offset.

In [21] detected lights are paired by using cross-correlation where color pixels are used directly thus avoiding sensitivity to a thresholded image. Threshold for correlation is derived by fitting Gaussian onto histogram of cross-correlation values.

$$\gamma = \sum_{x,y} \frac{(T(x, y) - \bar{T})(I(x, y) - \bar{I})}{\sigma_T \sigma_I}$$

(6)

where $\bar{T}$ and $\bar{I}$ are the mean values and $\sigma_T$ and $\sigma_I$ are the standard deviations of an image $I$ and template $T$ values respectively.

During tracking, the initial pairing threshold is relaxed. To improve robustness against blob symmetry disruption, only one blob should meet the similarity between frames.

The [15] and [16] are using simple clustering where found blobs are then recursively clustered according to the following rules: same class label, symmetry, the ratio between width and height of the bounding box and vertical projection overlap.

Pairing is exploited in [5] where detected blobs are paired by using the rules that are based on geometric properties: area similarity, vertical coordinate similarity, bounding box width interval, width / height ratio and correlation between blobs (see Figure 5).

The between frame object pairing is done in predicted region where the number of nuisance bright spots is small. An adaptive thresholding is proposed where thresholds which are used to pair vehicle lights are adopted according to the size change of the tracked object.

Paper [14] uses top down clustering approach meaning that all blob combinations are included therefore $1/2 \cdot n \cdot (n-1)$ possible pairs are generated and then by applying the following rules false positive candidates are eliminated: blobs are in the same horizontal line, blobs have similar shapes and area and they have similar type (HS probability for headlight, tail-light and blinker).

Dissimilarity values are normalized and the total dissimilarity measurement is obtained by summing those values.

**C. Classification**

An SVM is used in [13] and [4] to classify every potential light object as headlight or non-vehicle light. An object is considered valid if it is classified as headlight during few frames (5 selected). For the negative samples they consider illuminated traffic signs and possible reflections or nuisance artefacts. Used features are common binarized blob features and black hat score is obtained by intersecting ellipses that are fitted on the actual spot position and the vertical similarity are used to reduce false positives. All factors are multiplied and compared with predefined threshold. In [19], a total of five motions features are extracted: horizontal and vertical speed magnitude, direction and motion displacement.

![Figure 5: Blob intersection which uses cross-correlation where score is obtained by intersecting ellipses that are fitted on binarized light blobs. $C_x$ is absolute correlation matrix for blob $x$. $S_{COV}$ is normalized similarity magnitude. The similarity is used as one of features in [2]. Left: Same vehicle light comparison. Right: Lights comparison from two different vehicles.](image-url)

Particle filter is also used for blob detection and pairing. In [27] blob detection is done by using multiple particle filters. Each particle has a weighting function that consists of:

1) edge activity - the normalized magnitude in a region of interest
2) horizontal symmetry position - a position is searched which has the maximum ratio between edge activity and symmetry error function which is normalized difference between magnitudes
3) the number of circle pairs - a circle is detected in the following way: the input image is binarized and the edge detection is applied. Gradient orientations are pairwise compared according to their position. The result is accumulator image of potential circle candidates [29].

The number of circle pairs is determined by taking the minimum number of votes for the left and right bounding box which correspond to the right and left vehicle light. The weighting functions are calculated by using those features. Those weights are assumed to be independent so they are simply multiplied.

Paper [20] uses features that are extracted from binarized blobs and from image directly as in [V-A]. Classification is done by semi-automatic parameter adjustment where parameters are roughly selected in a manual way and adjusted by Structured diferential learning technique [30].

RealAdaBoost is used in [22] to classify blobs. There are 4 groups of features that are used for classification:

1) binary - like area, centroid, elongation (does not write all features)
2) features that describe intensity in monochrome image (maximum, mean, max position...)
3) the same as 2) just for red channel
4) color information features

A mutual classifier training is used where counter-samples for one classifier are used as samples for another classifier because discriminative classifiers generalize better by using samples and counter-samples around the discriminative border [22].

If the classifier output is below some threshold additional classification is made where pairing blob is searched. Features that are used for pairing are ratio of levels, distance of blobs. Those features are fed into RealAdaboost algorithm. Total confidence is promoted to upper possibility if the pairing classifier output is true.

In [19] SVM and AdaBoost classifiers are trained and their performance is compared. There is no object pairing. There are four types of classifiers: head-light, tail-light, street-light and other object classifier.

SVM is trained by using the following features: position, brightness, shape, spatial relation features, color features and motion features.

Adaboost features are: 5 Haar wavelet features, hue and centroid. Weak classifier in the Adaboost algorithm is a tree. The paper concludes that SVM has more stable behaviour - the light switching does not occur frequently, while AdaBoost has smaller error, but tends to switch lights more often.

According to Gentex patent [23] blob classification is done by applying the rules in Table I

### VI. Tracking

A tracking phase is often interleaved with detection, pairing and classification. Some systems do not require tracking because their output is based only on the current frame. On the other hand blob tracking is the base task for systems that make output based on object motion history. Therefore, blob tracking procedure must be robust enough to deal with temporary occlusions (e.g. one vehicle occluding another vehicle during overtaking). In this paragraph several approaches are described where some of them are developed specifically for vehicle blob tracking.

#### A. Kalman filter

In [13] and [4] after the initial detection, Kalman filter estimation is used to find the closest matching object from a previous frame, where the state of the Kalman filter consists of objects’ two-dimensional coordinates.

If an object is detected at a distance which is twice as farther than any distance before the object is classified as preceding, otherwise it is classified as oncoming.

In [21] and [26] Kalman filter is also used for blobs tracking.

#### B. Postprocessing - temporal smoothing

In [20] there is no tracking system. Instead a temporal smoothing is used where the headlight state is not switched if less than predefined number of frames a different classifier output is made. This approach smoothes noise. A hysteresis is applied meaning that different threshold is used to switch the headlight state again. The first threshold has a greater value than others thus the initial conditions are relaxed.

#### C. Rule based tracking

The [17] and [15] applies only rules and the paper [16] extends tracking by actions. Blobs are continuously merged and split. The actions defined in Table II are applied.

#### D. Autoregression


\[
X_t = \sum_{i=1}^{p} X_{t-i} + \epsilon_t
\]

where \(X_t\) is the state vector in a time frame, \(p\) is order of a model and \(\epsilon_t\) is the measurement error.

Five previous positions are used to predict a new position. AR model of order 3 is used. The advantage over Kalman filter

<table>
<thead>
<tr>
<th>Blob type</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>extremely bright light</td>
<td>average intensity is greater than threshold</td>
</tr>
<tr>
<td>AC light source</td>
<td>8 images in a small ROI are taken at high frame rate (&gt; 400 fps) and the AC magnitude is calculated. The idea behind this is that AC lights pulsate, and this can be detected at high frame rate.</td>
</tr>
<tr>
<td>traffic sign</td>
<td>object size is greater than vehicle light threshold and its average intensity is above threshold for traffic sign intensity</td>
</tr>
<tr>
<td>tail-light</td>
<td>three conditions are checked: non AC light, red / cyan ratio and average intensity which must be greater than tail-light brightness threshold</td>
</tr>
<tr>
<td>headlight</td>
<td>three components are checked: non AC light, red / cyan ratio and average intensity which must be greater than headlight brightness threshold - there are two thresholds (higher and lower) for bright and less bright vehicle head lights</td>
</tr>
</tbody>
</table>
is the absence of motion model, and the particle-filter has poor real-time performance.

In [19] a tracking procedure is mentioned but no detailed explanation is given. If the number of detected blobs exceeds a specified amount or if the ambient brightness value is strong enough the low beam state will be chosen. However decisions are not based on a single frame, but temporal smoothing is used to overcome possible errors which can be produced by detector output.

E. Nearest neighbour matching

In [14] confidence value of a tracked object is updated over time by a weighted average. Object tracking is performed by nearest neighbour update where a nearest blob is taken only if it is close enough. Merged lights from distinct vehicles and lights merged with blinkers are split by rising a threshold as described. When a detection of an already tracked object in a new frame is made, blob centroid and light shape over time are smoothed by 2D spline interpolation.

Paper [22] uses a spatial-temporal image which is consisted of vertical image projections that are accumulated over time. During vertical projection each pixel is weighted according to its location. The locations at which vehicles have a higher probability of being detected have greater weight value. Motion is modelled with probability and a tracked object state is determined by Hidden Markov Models -HMM.

In [23] the information about blobs are stored into memory. This is done only for bright blobs due to memory restrictions.

F. Particle filetring

In [27] blobs are tracked by using multiple particle filters. The first order motion model is assigned to each particle. To avoid locking more particle filters on the same object the hard gating is used which means that every filter sets restricted area for other filters. To solve problem of filter calculation priority several approaches are explored and the approach where filter has to obey to the restricted area of the other filter only if the other one is higher ranked (has better particles) is taken.

G. Other

In [2] successive blob detection, classification and pairing is used. Temporal information is stored into buffer of predefined size. Thresholds for pair miss-detection is also determined.

Paper [22] avoids tracking due to problem difficulty and instead it uses temporal coherence analysis. It is performed by using two matrices: the first contains hysteresis value called accumulation array, which range extends from 0 to a predefined number, and the second preserves actual state (true or false). Temporal smoothing is done by the following algorithm.

1) At the beginning all cells in accumulation array are set to zero. All cells in state array are set to false.
2) Accumulation array is decreased by a fixed number. Minimal value for each cell is zero.
3) To combine confidences from frame to frame accumulation and state array values are spread by using motion information.
4) The accumulation cells which correspond to the newly detected blobs are increased by a predefined amount.
5) Hysteresis is applied for the state array cells in the following way: if the accumulation array cell has a value that is greater than a half of the predefined maximum value state array cell is set to true. Array cell has to drop to 0 to set the corresponding state array cell back to false.

VII. PERFORMANCE MEASUREMENT

In [4] the following method is used to control vehicle’s high-beam. If the number of detected blobs is greater than a predefined threshold the headlights are automatically turned off. Headlights are detected at distances between 300 and 500 m and tail-lights at distances between 30 - 50 m. The detection rate reaches 100 percent for each light type and each tested sequence. The main problem which is stated is the time necessary to classify blobs correctly.

The method in [2] is used to detect time to collision in urban area. Methods has 95 percent detection rate with false positive rate below 9 percent for ego-lane. Detection rate and false positive rate are higher if all lanes are included. TTC deviation comparing to a radar as ground truth is below 15 percent and lower for close distances.

In [21] only rear-light detection and tracking is made. The reported detection rate of more than 94 percent is achieved for vehicles at distance of 0-50 m.

In [20] a system is made for high-beam dimming (not adaptive control). It is reported that the developed system works well in various conditions. ROC curve is not used but customer quality measure which consists of number of wrong light switches classified into several categories.

Other papers do not have quantitative results [18], [17], [16], [13].
In [19] the following measurements are incorporated to build false negative value - FNV: missing beam deactivation, late deactivation, early activation and false activation. False positive rate - FPV is built from corresponding opposite measurements. Total error is a weighted sum of FNV and FPV where factor that multiplies FNV is larger. In [14] rear-lights are detected and tracked for purpose of distance estimation. The number of false negatives is below 6 percent for urban areas and below 2 percent for highway.

The method in [20] mainly focuses on blob detection using auto exposure (AE) and low exposure (LE) fusion. It shows that AE-LE fusion is superior over fixed thresholding, adaptive thresholding and LoG in AE in terms of performance.

Particle filtering is used for detection and tracking in [27]. The method can track 5 vehicles with 10 particle filters in real-time. Other results are not given.

The method [22] uses machine learning and pairing criteria which is not mandatory. It is used for adaptive headlight control. The classification is over 90 percent for both vehicles and non-vehicle objects. For small tail-lights classification results are below 90 percent.

VIII. CONCLUSION

In this paper an overview of the state-of-the-art methods is given for on-road night-time vehicle detection and tracking. There is no general method for solving this problem although some patterns can be observed like light blob detection by using thresholds, or tracking which has been tried to solve mostly by using Kalman filtering. The feature that blobs come in pairs is heavily used. Classification is done by applying rules or by using machine learning approaches like SVM and AdaBoost. Camera vision based systems are proposed due to large field of view and to extract information that can not be obtained by using non-vision based system thus enabling to spot vehicle at far distances. Nevertheless there are many problems due to complex real-time conditions. Therefore vehicle night-time detection and tracking is still an open area with many potential for improvement.

REFERENCES


