A Method for On-road Night-time Vehicle Headlight Detection and Tracking

Darko Jurić and Sven Lončarić
Department of Electronic Systems and Information Processing
Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia
Email: {darko.juric, sven.loncaric}@fer.hr

***Abstract*—A novel method for vehicle detection and tracking, which is capable to correctly detect and track vehicle headlights in rural and urban areas is presented. The detection procedure does not require image thresholding, or other preprocessing, which is commonly used in other methods, thus offering robustness and performance. A tracking method based on Joint Probability Data Association Filter (JPDAF) efficiently associates object detections with existing tracks and provides a track management system. The method deals with object appearance and disappearance and can handle false detections introduced by the frame-based detector.**

***Keywords*—night-time vehicle detection, light blob detection, image segmentation, object tracking, object classification**

I. Introduction

Despite the fact that the traffic volume at night is much lower than during day-time, 42 % of all traffic accidents occur after dark and 58 % are fatal [1].

Monocular vision based systems that operate during the night rely on head-light and tail-light detection to spot and track vehicles. The most common systems built on top of the vehicle light detection and tracking are headlight control and forward collision warning. Driver distractions and unpredictable movements of preceding vehicles, especially during night, can lead to accidents. Although forward collision warning systems usually detect rear lights, front light detection is also important because the relative speed between two approaching vehicles is higher and those objects must be detected earlier. Vision-based systems can outperform radar based ones in terms of detection range and field of view. However, because the forward collision warning mainly operates in urban areas, vision-based systems have issues due to highly lit environment (street-lights and building lights).

This paper introduces a novel method, which is built on top of recently proposed methods for object classification - Markuš et al. [2] and tracking - Jurić et al. [3].

In Sec. II the current state-of-the-art methods for vehicle detection and tracking are grouped by category. Sec. III describes the proposed method. Sec. III-A and Sec. III-B present the main steps of the method - detection and tracking. In Sec. V experimental results are given. The conclusion is given in Sec. VI.

II. Previous work

The methods for vehicle light detection and tracking are divided into groups based on the common pipeline shown in Fig. 1. The common approaches for light detection are mentioned in Sec. II-A and for the tracking in Sec. II-B.

A. Light detection

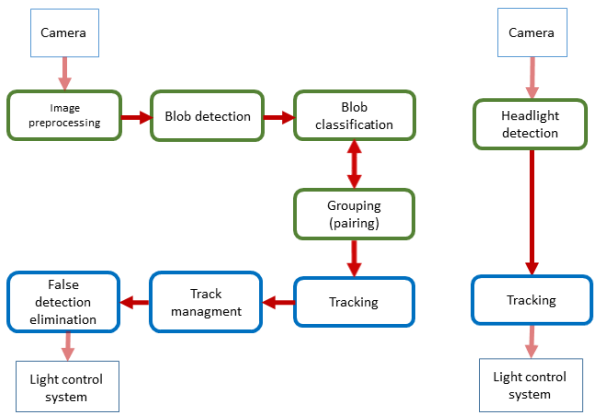
1) *Vehicle lights detection*: There are several approaches to lights detection. The most common approaches include fixed thresholding [4], [5] or adaptive thresholding [6]. A more advanced approach is multi-thresholding presented in [7]. Thresholding is fast but lacks the ability to capture objects' structure, which leads to LoG and LoG approximation filters [8]. In [9], a particle filtering approach is used to detect lights. The result of the stated methods are extracted blobs.
2) *Feature extraction*: Feature extraction is usually done after blobs are extracted and it is used for blob pairing and classification. Features are often calculated on already segmented images [10],[6], [7] and they include geometric blob properties. The most exploited feature calculated from an input image is the halo-effect [10].
3) *Lights pairing*: Lights pairing is often used to improve detection accuracy [11], [12]. However, in that case vehicles having only one light can not be detected. The method presented in [13] is based on a top down clustering approach. Among clustering-based methods cross correlation [14] and particle filtering [9] have been used.
4) *Classification*: Classification process is often based on rules and it is built on top of lights paring methods. The most used machine learning-based classifiers are support vector machine [6] and boosting [5].

B. Tracking

The tracking phase is often interleaved with detection, pairing and classification. The common used approach is the Kalman filter [10], [6]. A particle filtering-based method is presented in [9]. To reduce computational complexity an autoregressive model is used in [15]. Other methods utilise rule-based tracking [16], [11], [12] and nearest neighbour matching [13]. Since the lights tracking problem is hard the method presented in [5] uses temporal smoothing to build light switching system.

III. The proposed method

The state-of-the-art methods incorporate several stages as shown in Fig. 1. Our aim was to simplify the method and



(a) (b)

Fig. 1: Methods for vehicle light detection and tracking. 1a: The block diagram of a commonly used method. 1b: The block diagram of the proposed simplified method, which consists of only two stages.

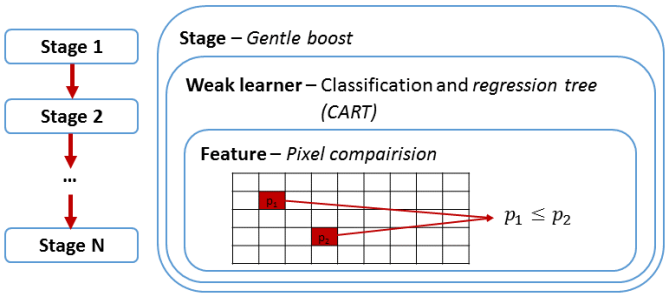


Fig. 2: The structure of the binary classifier used as vehicle headlight detector [2].

to improve detection and tracking robustness. The proposed method is comprised of two modules - detection and tracking module.

A. Headlight detection

The headlight detection method is based on a recently proposed classifier by Markuš et al. [2], which has several advantages:

1) The method merges image preprocessing, blob detection and classification steps into a single step.
2) The classifier operates on raw pixels eliminating the feature extraction step.
3) The detector has a computationally efficient cascade structure.
4) The method does not require parameter tuning except the threshold, which determines minimum confidence of the classified region.

The headlight detection method and the training procedure are described in the following text. The feature used for detection, as shown in Fig. 2, is the difference between intensities of a pixel pair called binary test Eq. (1).

$$\text{BINTEST}(I, l_1, l_2) = \begin{cases} 1, & I(l_1) \leq I(l_2) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

During the training process pixel pairs are chosen among randomly generated normalised pixel pair locations. The normalised pixel pairs can be easily rescaled to fit tested region. As the features are also point pairs they can be rotated to cover detection of rotated objects without any image manipulation or additional detector training. A weak learner, which is a regression tree, consists of internal nodes, which are binary tests and leafs representing the averaged class label outputs (+1, -1).

The whole region classification process can be described as an evaluation of the binary tests, getting regression trees outputs and summing them. The output is then compared against the stage threshold. This procedure is repeated for each stage until the final stage is passed, or it is terminated earlier if the value does not satisfy stage threshold in, which case the region is marked as a non-object region.

The input image is first converted to grayscale. Then the described binary classifier is executed over each image region. The multiple scale sliding window concept is used to scan an input image in order to detect vehicle headlights at different sizes and positions.

1) Detector training: The detector is trained in the following way. Vehicle headlights are annotated by bounding boxes. The boxes are randomly translated and scaled to enlarge training set. Each bounding box must have the same width-height proportion. To standardise the bounding box proportions the height scaling factor is calculated by obtaining all annotated rectangles and calculating the minimum scale for which all the lights are within the bounding box. The number of trees and the minimum true positive rate is set to a lower number in early stages to increase running performance and to enable the training procedure to discard samples, which are too complex to learn. This approach is also used in [2].

B. Tracking

The tracking procedure as shown is Fig. 3 is comprised of three nested parts. The core part is the extended Kalman filter [17], which is incorporated into Joint Probability Data Assocation Filter (JPDAF) [18]. The JPDAF method is extended to enable tracking the arbitrary number of objects [3]. Before a motion model can be incorporated in the Kalman filter a perspective transform coordinate mapping must be done. The whole tracking procedure is explained from its core parts - the perspective coordinate mapping to the extended JPDAF tracking management.

1) Perspective coordinate transformation: By observing a similarity between triangles that describe image and world coordinates the following relation can be written:

$$x = f \cdot \frac{X}{Z} \tag{2}$$

where X is the distance from a light blob of an oncoming vehicle to the camera lens centre, x is the distance from the light blob to the image plane centre. Z represents the distance

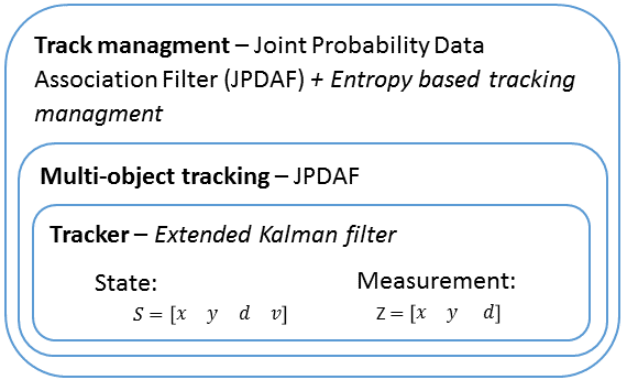


Fig. 3: The structure of the tracking algorithm. An object is tracked using the extended Kalman filter due to non-linearity of the motion model. Multiple object tracking is done by JPDAF.

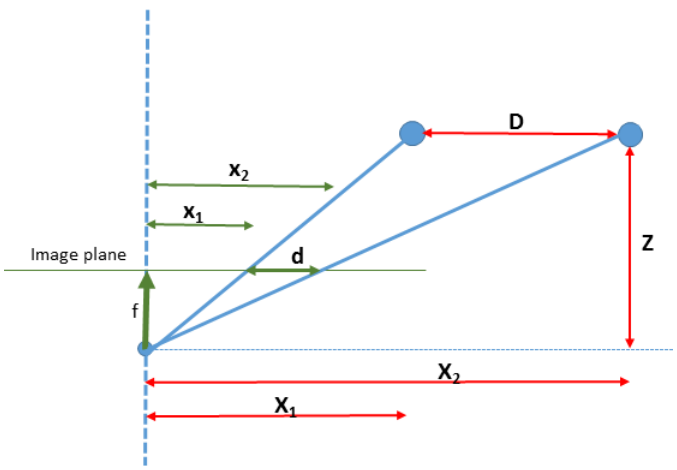


Fig. 4: The perspective model. Red color indicates world coordinates, while image plane coordinates are marked with green color.

from the camera to the oncoming vehicle. The focal length is marked with f. The world coordinates are expressed in meters, while the image measurements are in pixels. By expanding Eq. (2) for left and right vehicle light and differencing the two expressions we can obtain the distance between the two lights in pixels:

$$d = f \cdot \frac{D}{Z} \tag{3}$$

It is assumed that both vehicles move along the same line therefore X remains constant and Z changes:

$$x + \Delta x = f \cdot \frac{X}{Z + \Delta Z} \tag{4}$$

By inserting Eq. (2), Eq. (3) and $\Delta Z = v \cdot \Delta t$ into Eq. (4) the relation between $x(k)$ and $x(k-1)$ can be obtained:

$$x(k) = \frac{f \cdot D}{f \cdot D + v(k-1) \cdot d(k-1) \cdot \Delta t} \cdot x(k-1) \tag{5}$$

where k represents the time step. The expression for y coordinate update can be obtain analogously:

$$y(k) = \frac{f \cdot D}{f \cdot D + v(k-1) \cdot d(k-1) \cdot \Delta t} \cdot y(k-1) \tag{6}$$

In order to update x and y coordinate the update for distance d must be evaluated as well. The distance update is obtained by specifying Eq. (5) for x_1 and x_2 and differencing the obtained two expressions $d(k) = d + \Delta d = (x_2 + \Delta x_2) - (x_1 + \Delta x_1)$ as shown:

$$d(k) = \frac{f \cdot D}{f \cdot D + v(k-1) \cdot d(k-1) \cdot \Delta t} \cdot d(k-1) \tag{7}$$

The real world velocity can be obtained from Eq. (7):

$$v(k-1) = f \cdot D \cdot \frac{d(k-2) - d(k-1)}{d(k-2) \cdot d(k-1) \cdot \Delta t} \tag{8}$$

2) Extended Kalman filter: The state $S = [x\ y\ d\ v]$ of the Extended Kalman filter consists of detection coordinate, the distance between the lights in the image, and velocity. It is assumed that the relative velocity between the two approaching vehicles changes slowly, therefore the constant velocity model is used. The measurement vector is: $Z = [x\ y\ d]$. As the translation functions between world and image plane coordinates are not linear the Extended Kalman filter is used [17]. In order to eliminate the need to determine the focal length f and the vehicle light distance approximation D of the right side of Eq. (5) is divided by $f \cdot D$ to obtain a single parameter for estimation:

$$\frac{v(k-1)}{f \cdot D} \cdot p \tag{9}$$

where p is constant that converts pixels to meters. This is done analogously for Eq. (6) and Eq. (7).

3) JPDAF: This method [3] has several advantages:

1) Incorporates Kalman filter in a natural way to enable multiple-object tracking. Other methods do not provide track-management or they use extensions, which are developed only for specific cases like rules in [12].
2) JPDAF is a soft assignment technique meaning there is no strict detection-track assignment but the trackers are gradually updated by values obtained from association probabilities.
3) The false-positive handling is incorporated in the extended JPDAF by its design. There is no need for additional stage or specific operations to handle false-positives.

JPDAF procedure can be divided into three parts:

1) *Calculating $P(Z|X)$ matrix*: probability of a detection z_j given track x_t - $P(z_j|x_t)$, where the j and t are corresponding indices of detection and track respectively, is calculated for each detection-track pair. The $P(Z|X)$ matrix contains probabilities of the each pair. Probability for a pair is obtained by looking at Euclidean distance between the predicted position and the measurement. For each distance a

probability, which depends on type of a tracker, is assigned.

2) *Calculating association probability matrix - β*: all association hypothesis are generated. Each hypothesis contain possible associations. Each detection-track pair in a hypothesis is evaluated according to Eq. 10. The calculated values are multiplied. The obtained probability is added to the probability of detection-track pairs in hypothesis.
3) *Updating tracker*: the update procedure is tracker-dependent and it is described in [3]

If hypotheses regarding association θ are known, the probability of knowing detection z_j is calculated by Eq. (10).

$$P(z_j|\theta) = \begin{cases} P_F, & z_j \text{ false alarm} \\ P_D \cdot P(z_j|x_t), & z_j \text{ exisiting track} \end{cases} \tag{10}$$

where P_F is false alarm probability, and P_D is detection probability. Those values are constant as in the original method, however, they are initialised with TP and FP value obtained by the detector on the training samples. Although the standard Kalman filter is used in [3], the update equations remain the same for this case.

4) Tracking arbitrary number of objects: JPDAF can only be applied to scenes with a fixed number of objects. In order to support a variable number of objects, a Renyi entropy-based tracking management is proposed in [3]:

$$H_2(x_t) = -\log \int p(x_t)^2 dx_t \tag{11}$$

For Kalman filter a closed form solution of Eq. (11) exists as presented in [3] and is computationally inexpensive.

In the method [3] detection-track association depends on track association probability defined by

$$\beta^t = \sum_{j=1}^{M} \beta_j^t \tag{12}$$

where M is the number of measurements and t is the track index. In this paper a new tracker is created and associated with appropriate detection only if the track association probability shown in Eq. (12) is bellow a threshold.

When newly created tracks, marked as *tentative*, are associated with appropriate detections their initial entropies are stored as the reference entropies. If the entropy rises by a specified amount above the initial entropy the track is removed. If the object is tracked continuously the entropy will drop bellow the initial value. The track is labelled as *confirmed* if the drop amount is larger than a user-specified threshold.

The thresholds are fixed for all tracks. In order to set the entropy threshold value for track removal a compensation has to be made. If the threshold value is set too close to the initial entropy the existing tracks could be reinitialised. If the value is too high the removal of false positives would last too long. However, false-positives are usually not detected continuously meaning that a wider selection range for the upper entropy threshold value can be used.

IV. Parameter selection

The proposed method has been implemented and experimentally evaluated with a choice of parameters presented in this section. The most important parameter is the detection confidence threshold, which is set to fixed value of 2.5. The bounding box scale range goes from [11 x 5] up to [220 x 100] where the scale factor is 1.1. The bounding box width-height ratio is 2.2.

The tracking procedure has four parameters regarding track management and three parameters for the Extended Kalman filter. The entropy for the track removal is set to 1.15 of the initial entropy threshold value and the minimum entropy threshold value, which is used to confirm track, is set to 0.85 relative to the initial entropy. The minimum β^t value in Eq. (12) is set to 0.1. Kalman process noise Q and measurement noise R must be determined in advance. To measure the process noise several tracks are manually annotated. The constant velocity is assumed. Each obtained curve for the horizontal position x of the annotated tracks are fitted to the process model where the value in Eq. 9 is set to minimise error measured using the Euclidean norm. It is ensured that the reference and the fitted curve have the same starting and ending value. The standard deviation is taken for each fit. The average deviation is taken as the process noise for the x. The procedure is repeated for the vertical position and vehicle light distance. The velocity noise is determined empirically. The measurement noise is measured using annotated samples and grouped detector outputs. The standard deviation between detection and annotation bounding boxes is taken for its position and width. The initial velocity magnitude can be roughly determined and as it is adjusted by the Kalman filter.

V. Experimental results

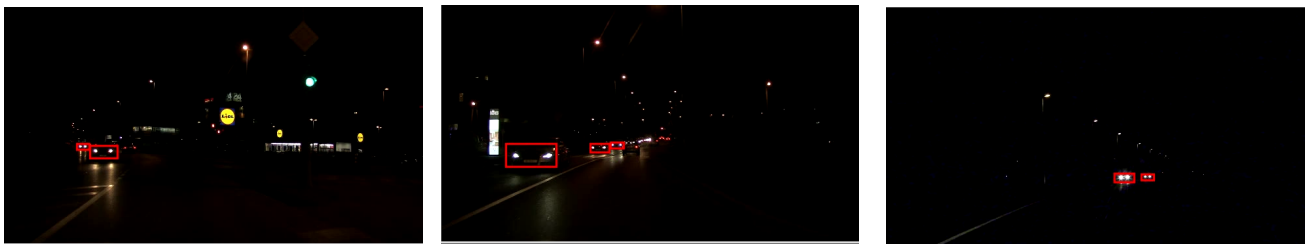
To demonstrate robustness of the detector to missed detections and false positives test video sequences have been manually labelled. Video resolution is 1280 x 720 but they are scaled to 640 x 360 when testing was conducted. The detection algorithm is evaluated frame by frame by measuring bounding box intersection between annotation and the bounding box obtained by grouping a detection. If the intersection percent is more than 70% the detection is proclaimed as valid. The results of these experiments are summarised in Table I. The sample test scenes are shown in Fig. 5 and the Fig. 7 shows some incorrectly classified samples.

Scenario	Correct Detections	Missed Detections	False Positives
Rural	834	52	14
Urban	1268	110	56

TABLE I: Detection results.

The missed detections shown in Table I are mostly result of excessive glare of closer vehicles. Farther vehicles, which can be hardly recognised by its lights also contribute to the missed detections. The false positives can be resolved by training the classifier with more samples and by incorporating the knowledge about the object coordinates.

To evaluate tracking performance the number of tracks, which are labelled as *confirmed* track is counted. Tracking



(a) (b) (c)

Fig. 5: Various environments, which served for the method testing. On 5a and 5b it can be noticed various nuisance lights that must be discarded. Image 5c shows that this approach is also suitable for rural environment where early detection is needed.

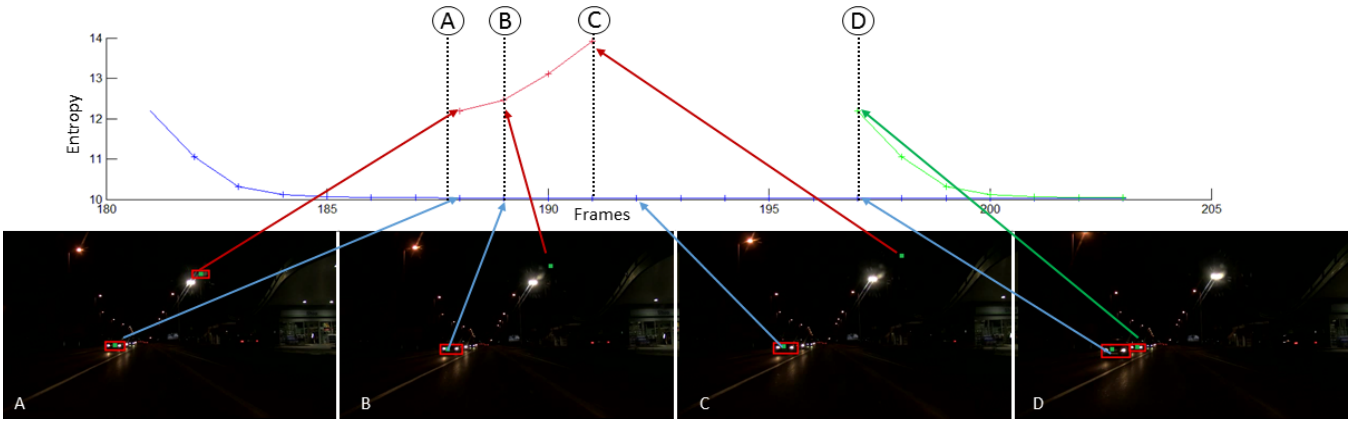


Fig. 6: The tracking algorithm can successfully handle false positive detection, which is shown on the four images above. Image A shows the appearance of the false positive detection and the initialisation of the Kalman tracker. On the image B the entropy of the Kalman tracker is increasing as the false positive does not have continuous appearance. The image C shows the tracker state and the entropy after 5 frames. Finally on the image D entropy has breached its maximum and the tracker is destroyed.

(a) (b) (c)

Fig. 7: To avoid creating the impression of perfect detection results we also depicted some of the problems that can occur. On 7a false positive is found between two car lights. Image 7b shows false detections on road reflection. The coordinates of detected objects are not taken into consideration, which can lead to false detection of similar looking objects (urban lighting) as shown in 7c.

is evaluated on obtained detections, which are grouped. The groups that have less than three detections are automatically discarded.

Scenario	Correct Tracks	Missed Tracks	FP Tracks
Rural	11 (92 %)*	1	1
Urban	45 (87 %)*	2	3

TABLE II: Tracking results where ”*” denotes the average percentage of correct successful tracking per track.

In Table II it can be seen that the tracking method successfully associates detections with the existing tracks. Resolving false positives is shown in Fig. 6. However, some false-positive detections can not be resolved: urban lighting and some road reflection detections, which are present during a few consecutive frames, so the tracker labels the track as *confirmed track*.

A direct comparison of the proposed method to other methods using the same video data would not be relevant as the video data used by other authors is not publicly available. In addition, some methods do not provide parameters needed for their exact implementation.

VI. Conclusion

In this paper a novel approach for vehicle headlight detection and tracking is proposed, which substantially differs from the previously published methods. The main advantages of the proposed method are: architecture simplicity compared to the standard detection and tracking pipeline, simplicity and robustness of the detection method, and the advanced tracking method, which is able to track arbitrary number of objects and deal with false positive detections. Such detection and tracking methods are not found in the previous work. Experimental results have shown that the proposed method detects and tracks vehicle headlights with a high accuracy.

The future work will include the following improvements:

- False positives (street lighting) could be eliminated by incorporating the knowledge about the object position and trajectory.
- During the detection phase a whole image is scanned with windows in all scales - no prior knowledge about a vehicle is incorporated, which could result in significant speed-up of the detector.
- The method extension to track the rear vehicle lights.

Acknowledgment

This research has been partially supported by the European Union from the European Regional Development Fund by the project IPA2007/HR/16IPO/001-040514 VISTA - Computer Vision Innovations for Safe Traffic.

References

[1] *New Headlight Sensors Make Night Driving Safer*, Road and Travel Magazine, 2007. [Online]. Available: http://www.roadandtravel.com/autoadvice/2007/highbeams.htm

[2] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, “Object detection with pixel intensity comparisons organized in decision trees,” 2013. [Online]. Available: http://arxiv.org/abs/1305.4537

[3] S. Juric-Kavelj, I. Markovic, and I. Petrovic, “People tracking with heterogeneous sensors using jpdaf with entropy based track management,” in *Proceedings of the 5th European Conference on Mobile Robots (ECMR2011)*, 2011, pp. 31–36.

[4] Y. Li, N. Haas, and S. Pankanti, “Intelligent headlight control using learning-based approaches,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 722–727.

[5] A. López, J. Hilgenstock, A. Busse, R. Baldrich, F. Lumbreras, and J. Serrat, “Nighttime vehicle detection for intelligent headlight control,” in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2008, pp. 113–124.

[6] P. F. Alcantarilla, L. M. Bergasa, P. Jiménez, I. Parra, D. F. Llorca, M. Sotelo, and S. Mayoral, “Automatic lightbeam controller for driver assistance,” *Machine Vision and Applications*, vol. 22, no. 5, pp. 819–835, 2011.

[7] J. H. Connell, B. W. Herta, S. Pankanti, H. Hess, and S. Pliefke, “A fast and robust intelligent headlight controller for vehicles,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 703–708.

[8] R. Ogura and G. Ohashi, “Vehicles detection based on extremas in nighttime driving scene,” in *Consumer Electronics (I), 2012 IEEE*. IEEE, 2012, pp. 679–682.

[9] C. Idler, R. Schweiger, D. Paulus, M. Mahlisch, and W. Ritter, “Real-time vision based multi-target-tracking with particle filters in automotive applications,” in *Intelligent Vehicles Symposium, 2006 IEEE*. IEEE, 2006, pp. 188–193.

[10] P. Alcantarilla, L. Bergasa, P. Jimenez, M. Sotelo, I. Parra, D. Fernandez, and S. Mayoral, “Night time vehicle detection for driving assistance lightbeam controller,” in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 291–296.

[11] Y.-L. Chen and C.-Y. Chiang, “Embedded vision-based nighttime driver assistance system,” in *Computer Communication Control and Automation (3CA), 2010 International Symposium on*, vol. 2. IEEE, 2010, pp. 199–203.

[12] Y.-L. Chen, H.-H. Chiang, C.-Y. Chiang, C.-M. Liu, S.-M. Yuan, and J.-H. Wang, “A vision-based driver nighttime assistance and surveillance system based on intelligent image sensing techniques and a heterogamous dual-core embedded system architecture,” *Sensors*, vol. 12, no. 3, pp. 2373–2399, 2012.

[13] A. Fossati, P. Schónmann, and P. Fua, “Real-time vehicle tracking for driving assistance,” *Machine Vision and Applications*, vol. 22, no. 2, pp. 439–448, 2011.

[14] R. O’Malley, E. Jones, and M. Glavin, “Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 453–462, 2010.

[15] J. Wang, X. Sun, and J. Guo, “A region tracking-based vehicle detection algorithm in nighttime traffic scenes,” *Sensors*, vol. 13, no. 12, pp. 16 474–16 493, 2013.

[16] Y.-L. Chen, C.-T. Lin, C.-J. Fan, C.-M. Hsieh, and B.-F. Wu, “Vision-based nighttime vehicle detection and range estimation for driver assistance,” in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 2988–2993.

[17] M. I. Ribeiro, “Kalman and extended kalman filters: Concept, derivation and properties,” *Institute for Systems and Robotics*, p. 43, 2004.

[18] Y. Bar-Shalom, “Extension of the probabilistic data association filter in multi-target tracking,” in *Proc. 5th Symp. on Nonlinear Estimation*, 1974, pp. 16–21.