VEHICLE DETECTION FOR NIGHTTIME USING MONOCULAR IR CAMERA WITH DISCRIMINATELY TRAINED MIXTURE OF DEFORMABLE PART MODELS

Hossein Tehrani Nik Nejad, Taiki Kawano, Seiichi Mita

Smart Vehicle Research Center, Toyota Technological Institute, Nagoya

Received September 30, 2011

ABSTRACT

Vehicle detection at night time is a challenging problem due to low visibility and light distortion caused by motion and illumination in urban environments. This paper presents a method based on the deformable object model for detecting and classifying vehicles using monocular infra-red camera. In proposed method, features of vehicles are learned as a deformable object model through the combination of a latent support vector machine (LSVM) and histograms of oriented gradients (HOG). The proposed detection algorithm is flexible enough in detecting various types and orientations of vehicles as it can effectively integrate both global and local information of vehicle textures and shapes. Experimental results prove the effectiveness of the algorithm for detecting close and medium range vehicles in urban scenes at night time.

Keywords: Pattern Recognition, Vehicle Detection, IR Camera, Histogram of Oriented Gradients, Support vector Machine.

1. INTRODUCTION

Collision prevention, especially at night time, is a crucial safety requirement in developing new vehicles. Many automobile manufacturers are trying to enhance the night vision by developing advanced headlight and safety systems. Existing work in the field of vehicle detection can be differentiated mainly by the type of sensors used [1]. Until now, vehicle detection at night is only successful when using expensive sensor systems. In the last years, the detection of vehicles using monocular video sensors, even at night, has attracted more attention. Initiated by the research on algorithms for traffic surveillance cameras, Cucchiara presented a system for night-time detection of vehicles using morphological operators, only by their position lights [2]. However, when using fixed camera positions, image regions can be masked out to prevent detection failures; thus, this method cannot be used by on-road vehicle systems.

Even in darkness, in cases where vehicle contours are still visible, edge detection algorithms may be used to find possible vehicle candidates [3]. After a potential candidate is detected, a color-based classification is used to detect the various light types, like front and rear position, break, or turn indicator lights. To detect vehicles by their position lights only, binary threshold filters can be used [4]. However, these methods are prone to illumination changes. More promising solutions for light blob detection are available; for example the morphological top-hat operator used in the video based detection of traffic lights [5]. Besides the detection of position lights, clustering a pair of lights the clustering of two light pairs is essential to form proper hypotheses. The correct association of light pairs is usually achieved using rule-based methods [6], or symmetry [2]. The existing methods presented methods for vehicle detection at night mainly use light blob features, such as headlight or taillight, for vehicle detection and classification. In the urban environment, there are many similar light blob features that could be mistakenly detected as a vehicle.

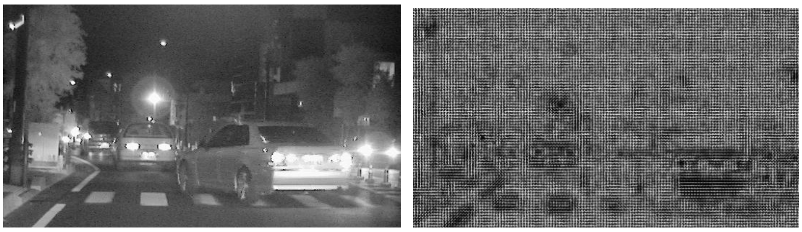
The presented methods, based on light blob features, cannot be used to detect parked vehicles. In urban roads, there are many streetlights, strong ambient illumination and unshielded bulkhead lightening, which can easily interfere with a vehicle’s head or taillights. Light blob detection methods are vulnerable since vehicle light blobs vary in their appearance (such as bumper lights, extra HID light bulb, etc.). In this paper, we present a method, which considers more vehicle features for night-time detection compared to only light blob features. We used a deformable object model, which showed high pattern recognition ability in the 2008 PASCAL Visual Object Classes Challenge [7]; the research results of one of the authors was used in this paper [8]. The object detection algorithm is flexible enough in detecting various types of vehicles as it can effectively integrate both global and local information of vehicle textures and shapes. The deformable objects model has already produced state of art results for detecting and tracking vehicles at day time [8]. Though at night time the visibility of the vehicles on the road is limited, there are still some features, such as headlights and taillights, which are more visible for oncoming and preceding vehicles. The main contribution of this paper is to use and adapt deformable part model for vehicle detection at night time.

We define the deformable part model of the vehicle and try to learn the parameters through an enormous number of positive and negative samples. We use the latent SVM and stochastic gradient method to learn the model parameters, and a data mining algorithm that allows for learning in very large datasets. The proposed method can be used in on-road systems to detect vehicles and in complex environment, such as urban roads, to detect moving and parked vehicles.

2. DEFORMABLE PART MODEL

Robust vehicle detection method for practical driving at night time in urban scenes should satisfy the following conditions:

1. Detection ability for various sizes and aspect ratios
2. Reliable in illumination, light distortion and light flood at night time at urban environment.
3. Detection ability of plural vehicle directions (front, rear, side)
4. Low false positive rate.



(a) Original image (b) HOG feature map

Figure 1. HOG representation of the image

HOG features are robust for local shape variations. HOG features produce fewer false positives than Haar-like features in vehicle detection [9]. In this paper, the deformable object model-based detection algorithm is adopted for vehicle detection and generating detection scores map for the following advantages:

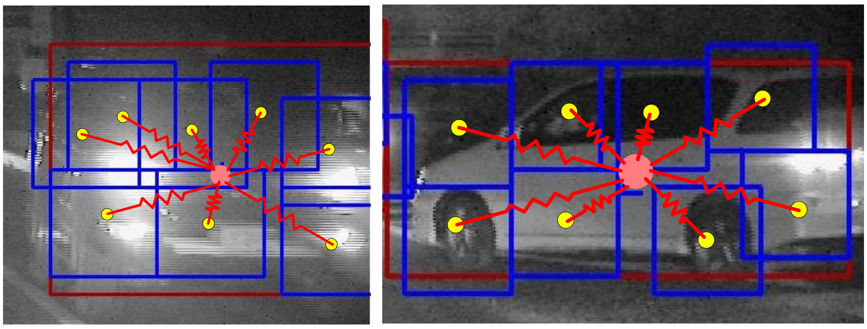
1. It can integrate global vehicle body information and local parts information such as wheels and windshield for increasing the efficiency and accuracy of detection.
2. It can be learned and used for detecting vehicle with different sizes and various aspect ratio vehicles.

2.1 Vehicle Model

All proposed models for vehicle detection involve linear filters that are applied to dense feature maps. A feature map is an array whose entries are d-dimensional feature vectors computed from a dense grid of locations in an image. In practice we use a variation of the HOG (Histogram Of Gradient) features from [10]. To calculate the HOG features, the input image is divided into many cells comprised of 8 × 8 pixels; each pixel votes for the orientation of its gradient with a strength that depends on the gradient magnitude for the cells as shown in Figure 1. A filter F , is a rectangular template defined by an array of d-dimensional weight vectors with sizes $\left[l_f, w_f\right]$. The response, or score, of a filter F at a position (x, y) of the feature map H , is the "dot product" of the filter and a subwindow of the feature map H with the top-left corner at (x, y).

$$\sum_{l_f, w_f} F\left[l_f, w_f\right].H\left[x+l_f, y+w_f\right] \tag{1}$$

A model for a vehicle with n parts is formally defined by a $n+2$-tuple, $\left(F_0, P_1, ..., P_n, b\right)$, where F_0 is a root filter with sizes $\left[l_f^0, w_f^0\right]$, P_i is a model for the i-th part and b is a real-valued bias term. Each part model is defined by a 3-tuple, $\left(F_i\left[l_f^i, w_f^i\right], v_i, d_i\right)$, where F_i is a filter for the i-th part with sizes $\left[l_f^i, w_f^i\right]$, v_i is a two-dimensional vector specifying an "anchor" position for part i relative to the root position, and d_i is a four dimensional vector specifying coefficients of a quadratic function defining a deformation cost for each possible placement of the part relative to the anchor position.



(a) Vehicle front model

(b) Vehicle back truncated model

Figure 2. Deformable vehicle detector model structure

2.2 Mixture Models for Vehicle

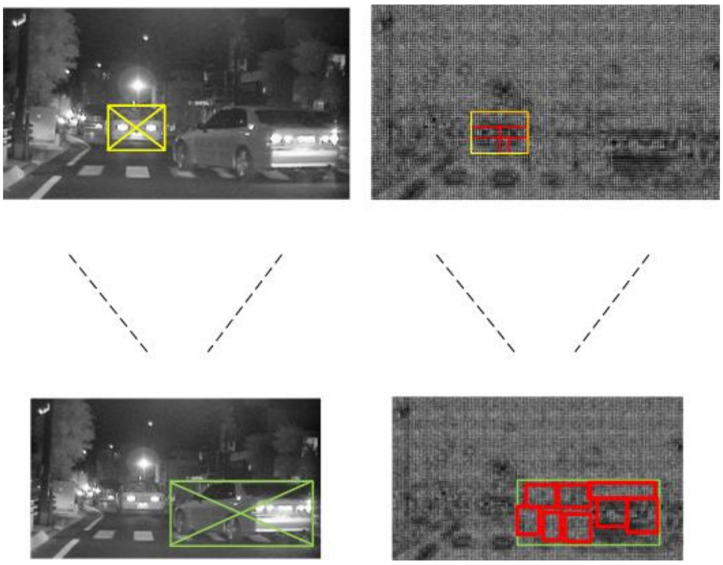
A mixture model with m components is defined by a m -tuple, $M = (M_1, ..., M_m)$, where M_c is the model for the c-th component of the vehicle. In this paper, we consider four components for the vehicle including front, back, front truncated and back truncated. Figure 2 shows the front and back truncated models. In Figure 2, the red rectangle represents the root filter, F_0, and the eight blue rectangles express part filters, $F_i, [i = 1, 2, ..., 8]$. The six springs connecting the root filter to each part filter are quadratic deformation cost functions.

2.3 Hypothesis Score Calculation

We would like to define a score at different positions and scales in an image. This is done using a feature pyramid, which specifies a feature map for a finite number of scales in a fixed range. In practice we generate feature pyramids by computing a standard image pyramid via repeated smoothing and subsampling, and then computing a feature map from each level of the image pyramid as shown in Figure 3. An object hypothesis specifies the location of each filter in the model in a feature pyramid, H, where $p_i = (x_i, y_i, l_i)$ specifies the level and position of the i -th filter.

Let $\phi(H, p, l \times w)$ denote the vector obtained by concatenating the HOG feature vectors in the $l \times w$ subwindow of H, with the top-left corner at p, in row-major order. The score of F_i at p is $F_i.\phi(H, p)$, obtained by concatenating the weight vectors in F_i in the $l_f^i \times w_f^i$ subwindow of H, with the top-left corner at p, in row-major order. We require that the level of each part is such that the feature map at that level was computed at twice the resolution of the root filter level.

The score of a hypothesis is given by the scores of each filter at their respective locations, minus a deformation cost that depends on the relative position of each part with respect to the root (the spatial prior), plus the bias.



(a) image resizing

(b) HOG pyramid

Figure 3. Procedure of HOG pyramid calculation. In the first level the far taxi has been detected and in the last level close gray sedan car has been detected in the HOG pyramid. The green and red boxes in the HOG feature map corresponds to the root filter and the red rectangles in the green box corresponds to the part filters

$$score(p_i)=\sum_{i=0}^{n}F_i.\phi(H,p_i)-\sum_{i=1}^{n}d_i.\phi_d(dx_i,dy_i)+b \tag{2}$$

where,

$$(dx_i,dy_i)=(x_i,y_i)-(2(x_0,y_0)+v_i) \tag{3}$$

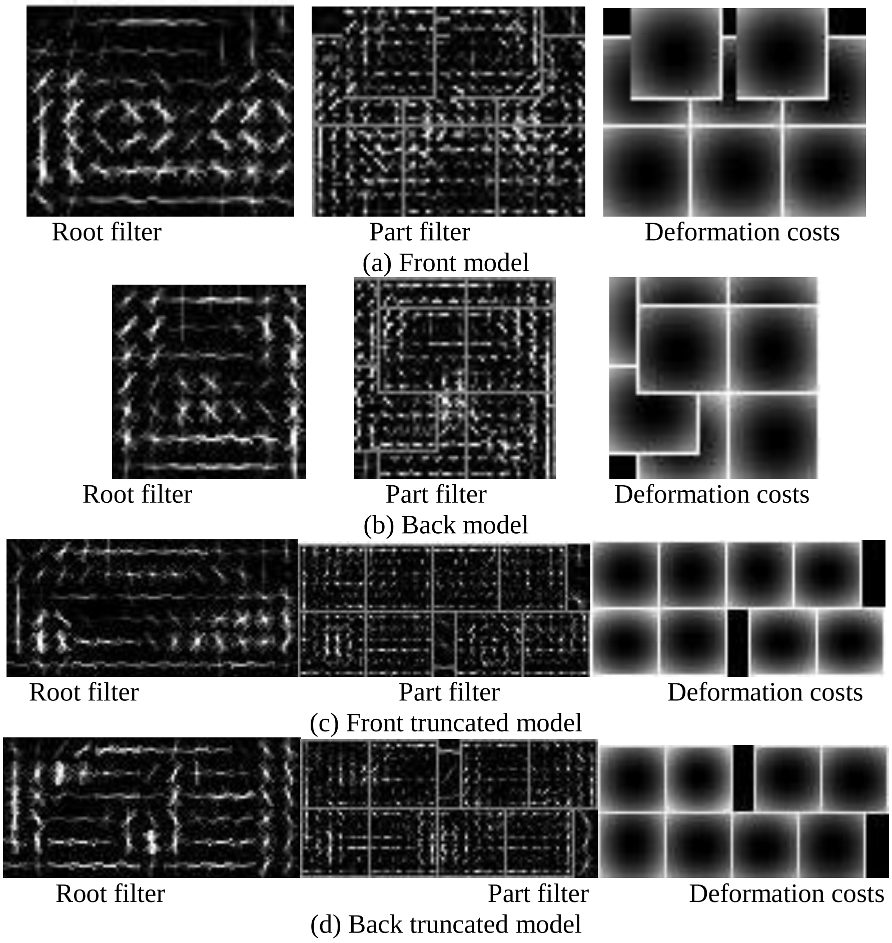
gives the displacement of the *i*- th part relative to its anchor position and

$$\phi_d(dx_i,dy_i)=(dx_i,dy_i,dx_i^2,dy_i^2) \tag{4}$$

are deformation features.

Note that if $d_i=(0,0,1,1)$, the deformation cost for the i-th part is the squared distance between its actual position and its anchor position relative to the root. In general the deformation cost is an arbitrary separable quadratic function of the displacements. The bias term is introduced in the score to make the scores of multiple vehicle models comparable when we combine them into a mixture model.

There exist a large (exponential) number of potential placements for a model in a HOG pyramid. We use dynamic programming and distance transforms techniques [8] to compute the best location for the parts of a model as a function of the root location. This computation takes $O(nk)$ time, where n is the number of parts in the model and k is the number of cells in the HOG pyramid. To detect objects in an image, we score the root locations according to the best possible placement of the parts and threshold this score.



Root filter Part filter Deformation costs

(a) Front model

Root filter Part filter Deformation costs

(b) Back model

Root filter Part filter Deformation costs

(c) Front truncated model

Root filter Part filter Deformation costs

(d) Back truncated model

Figure 4. Trained parameters for four component vehicle models at nigh time

An object hypothesis for a mixture model specifies a mixture component, $1 \le c \le m$, and a location p_i for model M_c. The score of this object hypothesis is the score of the hypothesis for the c-th model component.

The score of a hypothesis z can be expressed in terms of a dot product, $\beta.\psi(H,z)$, between a vector of model parameters β and a vector $\psi(H,z)$,

$$\beta = (F_0,...,F_n,d_1,...,d_n,b) \tag{5}$$

$$\psi(H,z) = (\phi(H,p_0),...,\phi(H,p_n),\phi_d(dx_1,dy_1),...,\phi_d(dx_n,dy_n),1) \tag{6}$$

this relationship illustrates a connection between our models and linear classifiers. We use this relationship in the latent SVM framework to learn the model parameters.

3 MODEL TRAINING

We use the learning algorithm presented in [11] for training the parameters β. A latent

SVM is defined as follows. We assume that each example x is scored by a function of the form;

$$f_{\beta}(x) = max_{z \in Z(x)} \beta.\phi(x,z) \tag{7}$$

where, β is a vector of model parameters and z is a set of latent values. For our deformable models, we define $\phi(x,z) = \psi(H(x),z)$ so that $\beta.\phi(x,z)$ is the score of placing the model according to z.

In analogy to classical SVMs, we would like to train β from the labeled examples $D = \left(\langle x_1, y_1 \rangle, ..., \langle x_n, y_n \rangle\right)$, where $y_i \in \{-1,1\}$, by minimizing the following objective function,

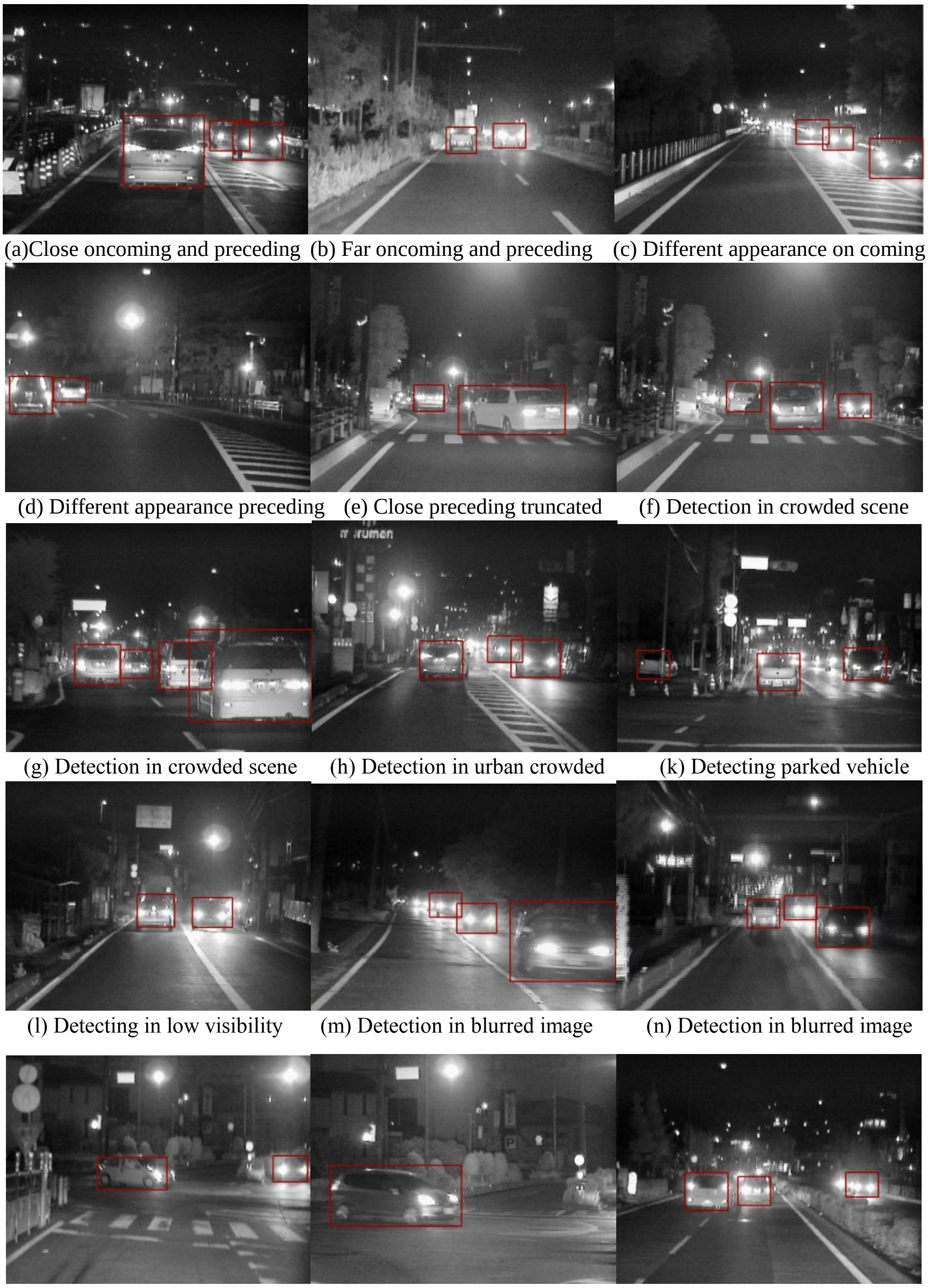
$$L_D(\beta) = \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i . f_{\beta}(x_i)), \tag{8}$$

where $\max(0,1-y_i . f_{\beta}(x_i))$ is the standard hinge loss and the constant C controls the relative weight of the regularization term. Note that if there is a single possible latent value for each example ($|Z(x_i)|=1$), then f_{β} is linear in β, and we obtain linear SVMs as a special case of latent SVMs. The results of training for filters of mixture models with four components are depicted in Figure 4.

4 EXPERIMENTAL RESULTS

The proposed method was applied and evaluated for practical detection and tracking in urban scenarios. The movies were generated in an urban environment by a vehicle-mounted monocular camera (Land cruse with close range IR monocular camera). The width and height of the movies were 640 (horizontal) and 480 pixels (vertical), respectively, and the frame rate was set to 30 fps. We trained four model components for close and medium range vehicles (less than 50 meter). Movies were evaluated for false positive FP (misdetection of non-vehicle regions), false negative FN (lost detection of vehicle regions) and true positive TP (correct detected vehicle regions) detections. For calculating TP, we only considered vehicles that totally appeared in the image, while for calculating FN we did not consider the occlusion FN.

A TP is considered when there is an overlap greater than 70% between the detected region D and the ground truth region G. Movies include various types of vehicles (sedans, wagons, and small cars) in Japan. Example results of the weighted deformable object model in various situations are shown in Figure 5 (detected vehicles are represented by rectangles). The proposed method detected various type and directions of vehicles for close and medium range, as shown in Figure 5 (a)–(q). The proposed method also correctly detects multiple vehicles (of varying size and type) in complex, congested intersection, scenarios. The proposed method could robustly detect parked vehicles as shown in Figure 5(k). Even light distortion due to motion blur did not affect the detection as it can be seen in Figure 5(m) and (n); however, detections depend on the distortion strength.



(a)Close oncoming and preceding (b) Far oncoming and preceding (c) Different appearance on coming

(d) Different appearance preceding (e) Close preceding truncated (f) Detection in crowded scene

(g) Detection in crowded scene (h) Detection in urban crowded (k) Detecting parked vehicle

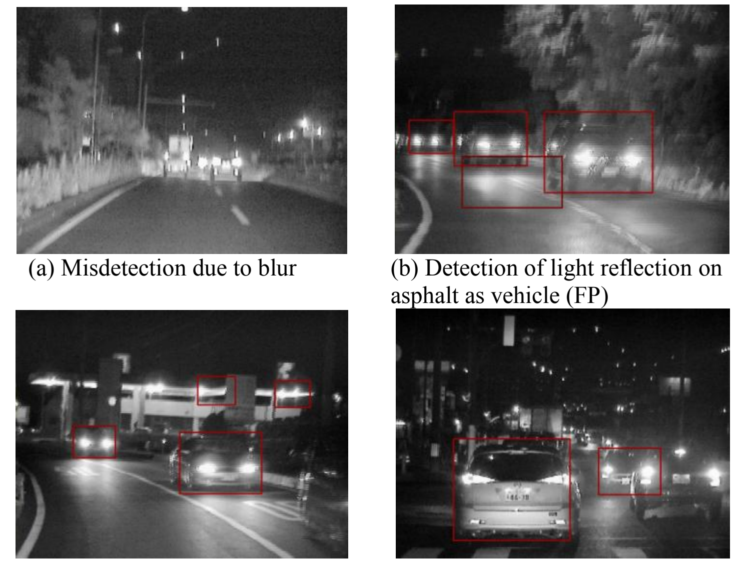
(l) Detecting in low visibility (m) Detection in blurred image (n) Detection in blurred image

(o) Detecting at intersections (p) Detecting truncated vehicle (q) Detection in illumination

Figure 5. Some correct detection results in different urban scenarios.

Table 1. Results of vehicle detection for proposed method

Scene	Frame No.	Ground Truth	Deformable object model					
			Detected vehicle	Rate %	False positive	Rate %	False negative	Rate %
I	3000	3981	2730	68.58	8	0.20	1250	31.40
II	1815	1148	845	73.60	1	0.08	302	26.30
III	3000	3120	1814	58.14	16	0.5	1304	41.79
Total	**7814**	**8249**	**5389**	**65.33**	**28**	**0.33**	**2856**	**34.62**



(a) Misdetection due to blur

(b) Detection of light reflection on asphalt as vehicle (FP)

(c) Misdetection due to occlusion

(d) False detection of light blobs of two vehicles at cross intersection as vehicle

*Figure 6***.** Failure results of the proposed method

Table 1 shows the evaluation results for the proposed method compared to the deformable object model. The proposed weighted deformable object model has correctly detected vehicles for more than 65.33% of the frames for all scenarios. In particular, the false positive rates in all scenarios were less than 0.33%, and there were only a few misdetections of non-vehicle regions. Thus, for the weighted deformable object model, the detection rate and false negative have improved, while the false positives have slightly increased. The primary reasons for false negatives include strong distortion due to vibration, occlusion, black vehicles and customized shape vehicles and lightening. Finally, some of the failure results of the proposed method are shown in Figure 6. False negatives tended to occur in strong distortion cases due to motion, as shown in Figure 6(a). An example of a false positive is shown in Figure 6(b) caused by the strong reflection of the vehicle headlights on the asphalt. In this paper we proposed a method for detecting the headlight reflection and removing it. Figure 6(c) shows a misdetection due to occlusion of two vehicles at an intersection. In this case, the two vehicles were detected as one because the deformable model incorrectly considered the taillight of the preceding vehicle as the headlight of the oncoming vehicle. This is a problem of windows based detection methods using deformable object models, which needs to be solved. Figure 6(d) shows false detections of light blobs as headlights of incoming vehicles.

5. CONCLUSION

This paper proposed a robust on-road multi vehicle detection method at nighttime for various practical driving scenes using an IR monocular camera. The proposed method is able to detect vehicles in low visibility conditions and complex situations at intersections in an urban environment. The presented method can detect different type and directions of vehicles in complex urban scenarios with illumination and light distortion due to motion blur. Some experimental results from practical urban scenarios showed that the proposed method can achieve an average vehicle detection rate of 65.33% with a low rate of false positives for complex environment such as urban roads. In the future, we are trying to expand the model to detect far range vehicles and improve the efficiency of the detection algorithm.

REFERENCES

1. Z. Sun, G. Bebis and R. Miller - On-road vehicle detection: a review, IEEE Trans. Pattern Recognition and Machine Intelligence **28** (5) (2006) 694-711.
2. R. Cucchiara and M. Piccardi - Vehicle Detection under Day and Night Illumination, In Proc. of IIA’99 - Third Int. ICSC Symp. on Intelligent Industrial Automation, Special Session on Vision Based Intelligent Systems for Surveillance and Traffic Control, 1999, pp. 789-794.
3. Cabani G. Toulminet, and A. Bensrhair - Color-based detection of vehicle lights, In Proc. IEEE Intelligent Vehicle Symposium (IV)., Jun 2005, pp. 278–283.
4. J. Firl, M. H. Hoerter, M. Lauer, and C. Stiller - Vehicle detection, classification and position estimation based on monocular video data, Automotive Lighting, Darmstadt, Sept. 2009.
5. R. de Charette, and F. Nashashibi - Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates, In Proc. IEEE Intelligent Vehicle Symposium (IV), Jun 2009, pp. 278–283.
6. T. H. Chen, J. L. Chen, C. H. Chen, and C. M. Chang - Vehicle detection and counting by using headlight information in the dark environment, IIHMSP 2007. Third International Conference on **2** (2007) 519–522.
7. Zisserman. PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. [online] Available. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.
8. P. Felzenszwalb, D. McAllester and D. Ramaman - A Discriminatively Trained, Multiscale, Deformable Part Model, Proceedings of the IEEE CVPR, 2008, pp. 1-8.
9. P. Negri, X. Clady, and L. Prevost - Benchmarking Haar and histogram of oriented gradients features applied to vehicle detection, Int. Conference on Informatics in Control 2007, May 2007, pp. 359-364.
10. N. Dalal and B. Triggs - Histograms of oriented gradients for human detection, Proc. of Computer Vision and Pattern Recognition (CVPR), Vol. 1, June 2005, pp. 886-893.
11. P. F. Felzenszwalb, D. McAllester, and D. Ramanan - Object detection with discriminatively trained part-based models, IEEE Pattern Analysis and Machine Intelligence (PAMI) **32** (9) (20101) 1627-1645.