Suspected Vehicle Detection for Driving Without License Plate Using Symmelets and Edge Connectivity

Jun-Wei Hsieh[1,2], Hung Chun Chen[2], Ping-Yang Chen[1], Shiao-Peng Huang[3]

[1] Institute of Com. Intelligence, National Yang Ming Chiao Tung University, Taiwan
[2] Dep. of Computer Sci. and Eng., National Taiwan Ocean University, Taiwan
[3] Chunghwa Tele. Lab., Dianyan Rd., Yangmei District, Taiwan
jwhsieh@nctu.edu.tw, tryitJimmy@hotmail.com, pingyang.cs08g@nctu.edu.tw, pone@cht.com.tw

Abstract

This paper proposes a novel suspected vehicle detection (SVD) system for detecting vehicles that are travelling on roads without a license plate. We start with detecting vehicles in a still image by utilizing a symmelet-based approach which allows us to determine a vehicle's region of interests (ROIs). A symmelet is a pair of an interest point and its corresponding symmetrical one. We modify the nonsymmetrical SURF descriptor into a symmetrical one in which no additional time complexity is added and no motion feature is required. This method allows for different symmelets to be efficiently extracted from road scenes. The set of symmelets can be used to locate the desired vehicle's ROI with the use of a projection technique. We then examine the existence of a license plate within this ROI, with an edge connectivity scheme that highlights possible character regions for plate detection. This SVD system provides two advantages; the background of the image does not need to be subtracted from analysis and the system does not require the use of a GPU. It is extremely efficient for real-time intelligent transport system (ITS) applications.

Keywords: Suspected vehicle detection, Symmelets, License plate detection

1 Introduction

Vehicle analysis is an important task in various applications, such as self-guided vehicles, driver assistance systems, electronic toll collection (ETC), crime prevention / investigation, intelligent parking systems, or in the measurement of traffic parameters such as vehicle count, speed, and flow. For the ETC system [1] in Taiwan, the task to detect moving violation for cars if their license plates are not hanged is typically important to maintain users' fairness and equality across the highway network. Basically, this ETC system is a distance-based and multi-lane free flow (MLFF) tolling system with 319 gantries to the road users. The average number of transactions per day is about 14 millions. Most transactions (94.14%) are achieved via a RFID tag (named as eTag in Taiwan). For the vehicles without using eTags, the system tries to find their owners by recognizing their license plates and then requiring them paying their tolls. Most recognition failures in this system are caused by the vehicles without hanging any license plates. Currently, this system lacks the ability to detect this tolling failures automatically and thus cannot intercept their forward passes in next gantry. In addition to ETC, for policemen in crime prevention and investigation, this SVD task is also important because the perpetrator in most of crime events will drive a car moving on road without any license plate.

A pre-requisite for enabling this analysis is to accurately locate vehicles in video images so that attribute extraction and comparison can be performed. In actual cases, there are significant variations in vehicles, including their colors, sizes, orientations, shapes, and poses. To treat this variation problem in vehicle appearance, different approaches [3-7] have been proposed that use different features and learning algorithms to locate vehicles. In the existing literature, most techniques [3-6] adopt background subtraction to detect moving vehicles from video sequences. For example, Faro *et al.* [3] used a background modeling technique to subtract possible vehicle pixels from the road and then applied a segmentation scheme to remove partial and full occlusions among vehicle blobs. In Unno *et al.* [4], motion information and symmetry properties were integrated to detect vehicles from videos. Jazayeri [5] used HMM to probabilistically model the vehicle motion and then distinguished the vehicles from the road. However, this type of motion feature is no longer usable and found from still images. Especially for the cases in Taiwan ETC system, only still images with a fixed high dimension are available to be analyzed due to the high accuracy requirement in license plate recognition.

Another commonly-adopted vehicle detection approach

*Corresponding Author: Jun-Wei Hsieh; E-mail: jwhsieh@nctu.edu.tw
DOI: 10.3966/160792642021032202021

involves training a robust vehicle detector through active learning [8-9]. For example, Chang and Cho [8] used an online Adaboost approach to train a cascade of strong classifiers to detect vehicles from a rear view. Sivaraman and Trivedi [9] used Haar features to train a strong vehicle detector with an Adaboost learning process consisting of off-line batch training, followed by a series of semi-supervised annotations and a batch retraining. This training-based method is effective at object detection but becomes unreliable for viewpoints that are not trained.

In addition to handcrafted features, in recent years, deep learning has been a very popular topic object detection. For example, in [10], Simonyan and A. Zisserman proposed a very deep CNN (convolutional networks) to build a VGG net for object detection and recognition. To improve the efficiency of VGG, Girshick et al. [11] proposed an RP-CNN architecture (Region Proposal-Convolutional Neural Network) to detect objects by pre-detecting embed candidates from the outside of the network. The YOLO system [12] can achieve detection rate with 45 fps by eliminating bounding box proposals and the subsequent feature resampling stage. Inspired by YOLO [12], the SSD method [13] used multiple layers of deep CNNs with lower resolution input to get state-of-the-art detection efficiency. Indeed, in terms of efficiency and accuracy, the above deep-learning approaches can achieve quite good performance in object detection. However, the requirement of GPU card in this deep-learning approach will increase huge cost if this re-identification system should be installed in many locations/gantries.

After vehicle detection, the next task in the SVD system is to detect the position of license plate. In the past, a number of techniques [14-24] have been proposed for locating the desired plate through visual image processing. The major features used for license plate detection include colors [15], corners [16], vertical edges [17], symmetry [18], projections of vertical and horizontal edges [20], and so on. For examples, Kim *et al.*[15] used color information and neural networks to extract license plates from images. However, color is not stable when the lighting conditions change. On the other hand, Dai *et al.* [20] used the projections of edges with different orientations to determine peaks of the histograms as possible locations of license plates. When the scene is complex, many unrelated edges will disturb the determination of the correct plate locations. Moreover, Yu and Kim [17] proposed a vertical edge-matching algorithm for grouping all possible positions of license plates through edge matching. In this approach, they assumed the vertical boundaries between a license plate and its backgrounds are strong. However, when the colors or intensities of license plates are similar to their backgrounds, the assumption is no longer valid. Other methods that use features like corners [16] and symmetries [18] also made the same assumption. In [21], a top-hat transformation is proposed to use mathematical morphological filters such as restraining noises, extracting features, segmenting images to efficiently enhance character regions for plate detection. Based on the top-hat transformation, in [22], an effective character-specific ER (Extremal Regions) method was proposed to refine the detected plate candidates. Furthermore, in [23], a 37-class convolutional neural network (CNN) is proposed and trained to perform a character-based plate detection using multiple scale sliding window. However, its efficiency is not good enough if without the aid of GPU

This paper develops a novel SVD system to detect vehicles on roads without a license palate using symmelets. Figure 1 shows the flowchart of our system. To initially achieve a highly accurate SVD system, the location of the vehicle region must be accurately detected. Without using any motion features, this paper proposes a symmelet-based detection scheme to detect vehicles from still images. A symmelet is a pair of an interest point and its corresponding symmetrical one. To efficiently build this symmelet feature, a fast SURF-like feature point extraction method is proposed to detect possible SURF-like feature points without using any GPU. After that, this paper modifies the non-symmetrical SURF descriptor into a symmetrical one without adding any time complexity. Then, different symmelets can be very efficiently extracted from road scenes. The central line of a vehicle can be detected from the set of extracted symmelets through a projection technique. Even with only one still image, each desired vehicle ROI can be accurately located along this symmetrical line. To ensure whether a license plate exists within the ROI, a novel scheme is proposed to highlight character regions for plate detection based on their edge connectivity. Then, the plate candidates can be finally verified to ensure whether a real license plate exists in the ROI based on a boosting method. Without using any GPU, the SVM system can detect each suspected vehicle extremely quickly even though frames with HD dimension are processed. As we know, this is the first framework to detect moving violation for vehicles which moves without any license plate. The major contributions of this work are noted as follows:

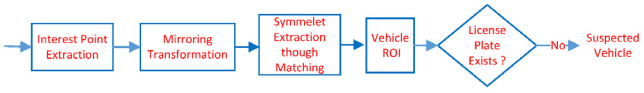


Figure 1. Flowchart of our approach for moving violation detection for vehicles moving without any license plates

(1) A fast SURF-like feature point extraction method is proposed to detect and translate a non-symmetrical SURF descriptor into a symmetrical one so that symmetrical objects can be very efficiently detected

from videos for various applications.

(2) A new symmelet-based vehicle detection scheme is proposed to detect vehicles from moving cameras. The advantage of this scheme is no need of background modeling and subtraction. In addition, it is suitable for real-time applications because of its extreme efficiency.

(3) An edge-connectivity approach is proposed to detect license plates extremely quickly.

(4) This is the first framework to detect moving violation for vehicles moving without any license plate.

The remainder of this article is organized as follows. In Section 2, we present the detailed formulation of our symmelet descriptor. Section 3 discusses the details of vehicle detection. In Section 4, details of license plate detection are described. Extensive experimental analyses are reported in Section 5. The conclusions and future work are provided in Section 6.

2 Symmelets

SURF is a novel scale-invariant feature detector and descriptor for feature matching among different images. However, it lacks the ability to identify symmetrical pairs of feature points. Symmetrical properties exist in many man-made objects (furniture, vehicles, and buildings) and in natural scenes (the moon), natural objects (apples), or animals (faces). This section will represent a symmetrical transformation for object detection and recognition.

2.1 Interest Point Extraction

The SURF algorithm uses a Hessian matrix approximation on an integral image to locate interest points. Similar to SIFT, in order to maintain the scale-invariant property for feature matching, each interest point is detected by searching points with local maximums of filter responses across different scales. The search across different scales will reduce not only the efficiency of feature extraction but also the number of feature points to be matched. In fact, if the matching process is performed within the same image, the scaling-invariant property is no longer needed. In addition, after feature extraction, a voting technique is commonly required for object parameter estimation. Good and robust voting results can be gained if the set of feature points is dense. Based on the above considerations, we propose a novel interest point extraction method to extract dense feature points for feature matching.

In order to obtain dense matching pairs, this section seeks the local maxima of gradients as intensive interest points instead of using Hessian matrix (used in SURF point extraction). Let $|\nabla I(x,y)|$ denote the gradient magnitude of the image *I(x,y)*. This paper defines the feature point as the one whose edge response is the strongest with a local area; that is,

Condition A: *P(x,y)* is a local maxima of $|\nabla I(x,y)|$, *i.e.*, $|\nabla I(x,y)| = \max_{(x',y')\in N_p}\{|\nabla I(x',y')|\}$, where N_p is a neighborhood of *P(x,y)* within a window.

Let the dimension of each input image be $N \times N$. If N_p is a $M \times M$ window, the max operation in Condition A will be $O(M^2N^2)$ and becomes very time-consuming if M is large. Let $K = N/3$. To speed the scanning process, a smaller mask $K \times K$ is first used to find all possible candidates, *i.e.*, points with local maximums. By scanning I without overlapping, *i.e.*, stride= K, the process scans only $(N/K)^2$ blocks from which only the points with local maximums are recorded. At the second scanning, as shown in Figure 2, for each remained point, if it is the local maximum compared with its eight neighbor blocks, it is recorded as the SURF-like feature point. The time complexity for all scanning tasks is $O(N^2)$.

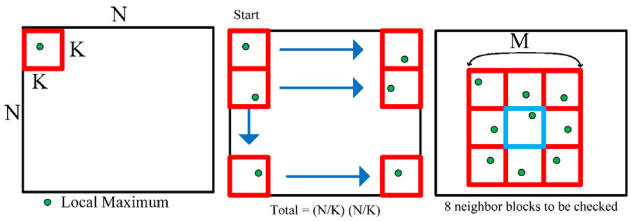


Figure 2. Fast SURF-like Feature Point Extraction

2.2 Symmetrical SURF Descriptor

The original SURF descriptor [2] lacks the support for matching two points if they are symmetrical. To provide this support, the relations of SURF descriptors between two symmetrical points must be derived [25]. Given a point P, its SURF descriptor is extracted from a 20×20 neighborhood of pixels around P. This neighborhood is further divided into 4×4 sub-regions B_{ij}. Then, distributions of gradient orientations and magnitudes are calculated and accumulated to form a 64-dimensional vector. Let B_o denote the original square extracted from an interest point and B_m be its horizontally mirrored version. For illustrative purposes, an 8×8 square example is shown as follows:

$$B_o = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} & b_{0,6} & b_{0,7} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} & b_{2,6} & b_{2,7} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & b_{3,6} & b_{3,7} \\ b_{4,0} & b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} & b_{4,7} \\ b_{5,0} & b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} & b_{5,7} \\ b_{6,0} & b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} & b_{6,7} \\ b_{7,0} & b_{7,1} & b_{7,2} & b_{7,3} & b_{7,4} & b_{7,5} & b_{7,6} & b_{7,7} \end{bmatrix}$$

and

$$B_m = \begin{bmatrix} b_{0,7} & b_{0,6} & b_{0,5} & b_{0,4} & b_{0,3} & b_{0,2} & b_{0,1} & b_{0,0} \\ b_{1,7} & b_{1,6} & b_{1,5} & b_{1,4} & b_{1,3} & b_{1,2} & b_{1,1} & b_{1,0} \\ b_{2,7} & b_{2,6} & b_{2,5} & b_{2,4} & b_{2,3} & b_{2,2} & b_{2,1} & b_{2,0} \\ b_{3,7} & b_{3,6} & b_{3,5} & b_{3,4} & b_{3,3} & b_{3,2} & b_{3,1} & b_{3,0} \\ b_{4,7} & b_{4,6} & b_{4,5} & b_{4,4} & b_{4,3} & b_{4,2} & b_{4,1} & b_{4,0} \\ b_{5,7} & b_{5,6} & b_{5,5} & b_{5,4} & b_{5,3} & b_{5,2} & b_{5,1} & b_{5,0} \\ b_{6,7} & b_{6,6} & b_{6,5} & b_{6,4} & b_{6,3} & b_{6,2} & b_{6,1} & b_{6,0} \\ b_{7,7} & b_{7,6} & b_{7,5} & b_{7,4} & b_{7,3} & b_{7,2} & b_{7,1} & b_{7,0} \end{bmatrix},$$

where $b_{y,x}$ is the intensity of B_o at the pixel (x, y). Let $B_{ij} = \begin{bmatrix} b_{2i,2j} b_{2i,2j+1} \\ b_{2i+1,2j} b_{2i+1,2j+1} \end{bmatrix}$ and $B_{ij}^m = \begin{bmatrix} b_{2i,2j+1} b_{2i,2j} \\ b_{2i+1,2j+1} b_{2i+1,2j} \end{bmatrix}$. We can then divide B_o and B_m into 4×4 sub-regions as follows:

$$B_O = \begin{bmatrix} B_{00} B_{01} B_{02} B_{03} \\ B_{10} B_{11} B_{12} B_{13} \\ B_{20} B_{21} B_{22} B_{23} \\ B_{30} B_{31} B_{32} B_{33} \end{bmatrix} \text{ and } B_m = \begin{bmatrix} B_{03}^m B_{02}^m B_{01}^m B_{00}^m \\ B_{13}^m B_{12}^m B_{11}^m B_{10}^m \\ B_{23}^m B_{22}^m B_{21}^m B_{20}^m \\ B_{33}^m B_{32}^m B_{31}^m B_{30}^m \end{bmatrix}. \quad \textbf{(1)}$$

For each sub-region B_{ij}, the sums of wavelet responses can be calculated by the form:

$$f_{ij} = (\sum_{b \in B_{ij}} dx(b), \sum_{b \in B_{ij}} dy(b), \sum_{b \in B_{ij}} |dx(b)|, \sum_{b \in B_{ij}} |dy(b)|), \quad \textbf{(2)}$$

where $dx(b) = b_{y,x+1} - b_{y,x}$ and $dy(b) = b_{y+1,x} - b_{y,x}$. For illustrative convenience, we use $d_{i,j}^x$, $d_{i,j}^y$, $|d_{i,j}^x|$, and $|d_{i,j}^y|$ to denote the four sums of wavelet responses in B_{ij}. Then, we have

$$d_{i,j}^x = b_{2i,2j+1} + b_{2i+1,2j+1} - b_{2i,2j} - b_{2i+1,2j},$$

and

$$d_{i,j}^y = b_{2i+1,2j} + b_{2i+1,2j+1} - b_{2i,2j} - b_{2i,2j+1}.$$

Let $B_i = \begin{bmatrix} B_{i0} B_{i1} B_{i2} B_{i3} \end{bmatrix}$ and $B_i^m = \begin{bmatrix} B_{i3}^m B_{i2}^m B_{i1}^m B_{i0}^m \end{bmatrix}$. From B_i, a new feature vector f_i can be constructed, *i.e.*,

$$\begin{aligned} f_i = (&d_{i,0}^x, d_{i,0}^y, |d_{i,0}^x|, |d_{i,0}^y|, d_{i,1}^x, d_{i,1}^y, |d_{i,1}^x|, |d_{i,1}^y|, \\ &d_{i,2}^x, d_{i,2}^y, |d_{i,2}^x|, |d_{i,2}^y|, d_{i,3}^x, d_{i,3}^y, |d_{i,3}^x|, |d_{i,3}^y|). \end{aligned} \quad \textbf{(3)}$$

Similarly, from B_i^m, another feature vector f_i^m can be constructed:

$$\begin{aligned} f_i^m = (&-d_{i,3}^x, d_{i,3}^y, |d_{i,3}^x|, |d_{i,3}^y|, -d_{i,2}^x, d_{i,2}^y, |d_{i,2}^x|, |d_{i,2}^y|, \\ &-d_{i,1}^x, d_{i,1}^y, |d_{i,1}^x|, |d_{i,1}^y|, -d_{i,0}^x, d_{i,0}^y, |d_{i,0}^x|, |d_{i,0}^y|). \end{aligned} \quad \textbf{(4)}$$

With f_i and f_i^m, the SURF descriptors $\boldsymbol{f}_o$ and $\boldsymbol{f}_{mir}$ of B_o and B_m can be constructed, respectively, as follows:

$$\boldsymbol{f}_o = [f_0 f_1 f_2 f_3]^t \text{ and } \boldsymbol{f}_{mir} = [f_0^m f_1^m f_2^m f_3^m]^t. \quad \textbf{(5)}$$

The transformation between $\boldsymbol{f}_o$ and $\boldsymbol{f}_{mir}$ can be easily built by converting each row f_i to f_i^m using the relations between Eqs. (3) and (4). Then, given two SURF descriptors $\boldsymbol{f}^p$ and $\boldsymbol{f}^q$, their distance is defined as

$$\xi_{SURF}(\boldsymbol{f}^p, \boldsymbol{f}^q) = \sum_{m=1}^{4} \sum_{n=1}^{16} [\boldsymbol{f}^p(m,n) - \boldsymbol{f}^q(m,n)]^2. \quad \textbf{(6)}$$

Given a frame I_t, we extract its set F_{I_t} of interest points according to the method described in Section 2.1. For an interest point p in F_{I_t}, we can extract its SURF descriptor $\boldsymbol{f}_o^p$ and obtain its mirrored version $\boldsymbol{f}_{mir}^p$ from Eq. (5). Let Min_p denote the minimum distance between p and other interest points in F_{I_t}, *i.e.*,

$$Min_p = \min_{s \in F_{I_t}, s \neq p} \xi_{SURF}(\boldsymbol{f}^p, \boldsymbol{f}^s). \quad \textbf{(7)}$$

In addition, let Min_p^{Sec} denote the distance of the best second match between p and other interest points in F_{I_t}. For another interest point q in F_{I_t}, if p and q form a symmetrical matching pair, they satisfy

$$\xi_{SURF}(\boldsymbol{f}_o^p, \boldsymbol{f}_{mir}^q) = Min_p \text{ and } \frac{Min_p}{Min_p^{Sec}} < 0.65. \quad \textbf{(8)}$$

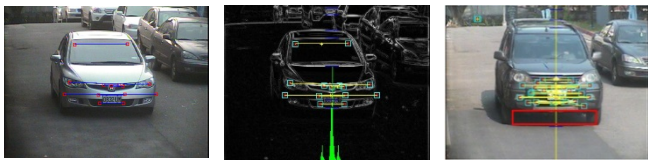
This paper names this symmetrical pair as a "**symmelet**".

3 Vehicle Detection

Vehicle detection is an important task for many applications such as navigation systems, driver assistance systems, intelligent parking systems, or the measurement of traffic parameters such as vehicle count, speed, and flow. To detect moving vehicles in video sequences, the most commonly adopted approach is to extract motion features through background subtraction. However, this technique is not stable when the background includes different environmental variations such as lighting changes, sun movements, or camera vibrations. In addition, motion features are of no use in still images. By taking advantage of "symmelets", a novel approach for detecting vehicles on roads without using any motion features will be proposed in this section.

After matching, a set of symmelets is extracted. Let $\mathbb{M}$ denote this set of symmelets and $c_{p \leftrightarrow q}$ denote the central position of one symmelet $\{p \leftrightarrow q\}$ in $\mathbb{M}$. With $c_{p \leftrightarrow q}$, a histogram-based method is therefore

proposed to determine the central line $L_{vehicle}$ of each vehicle candidate. The histogram (denoted by $H_{\mathbb{M}}$) is calculated by counting the x-coordinate of each central position $c_{p\leftrightarrow q}$ for all symmelts $\{p \leftrightarrow q\}$ in $\mathbb{M}$. The peak of $H_{\mathbb{M}}$ then corresponds to the central line of the desired vehicle candidate. Figure 3(b) shows the set of symmelets extracted from Figure 3(a). This histogram is plotted along the bottom of Figure 3(b). Figure 3(c) shows L_v denoted by a yellow line.



(a) (b) (c)

Figure 3. Matching results of symmetrical SURF points

After determining $L_{vehicle}$, a ROI R_v should be extracted to locate the vehicle candidate, *i.e.*, $R_v = (l_{R_v}, r_{R_v}, u_{R_v}, b_{R_v})$. Here, the symbols $l_{R_v}, r_{R_v}, u_{R_v}$, and b_{R_v} denote the left, right, upper, and bottom boundaries of R_v, respectively. In actual cases, a shadow region often exists underneath a vehicle. As shown in Figure 3(c), a shadow area (shown by a red rectangle) was found underneath the vehicle. The shadow region can be used to define this ROI and determine whether R_v is an actual vehicle. The area between the shadow line and the vehicle bumper will form a horizontal edge line l_{bumper} (as shown in Figure 4); it is used to define the bottom boundary of R_v, *i.e.*, $b_{R_v} = l_{bumper}$. l_{bumper} can be easily detected by using a horizontal Sobel edge detector. As for the upper boundary u_{R_v} (see Figure 4), because the hood and the vehicle window form a longer horizontal boundary, this paper uses this boundary to define u_{R_v}.

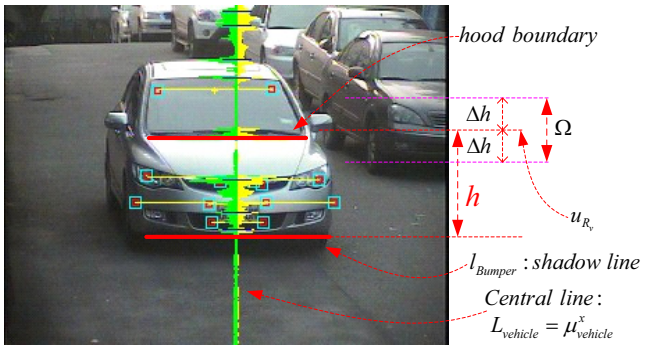


Figure 4. Notations for finding the upper and bottom boundaries of a vehicle

To extract the left and right boundaries of R_v, the vehicle width must first be defined. Let W_v and w_v denote its width in the real world and on the image plane, respectively. The relation between W_v and w_v can be derived as follows:

$$w_v(y) = -ay + b, \tag{9}$$

where a and b are the parameters used to build the relation between w_v and y. They can be easily estimated at the training state via a camera calibration technique. With $w_v(y)$, the left and right boundaries of R_v are defined, respectively, as follows:

$$l_{R_v} = \mu^x_{vehicle} - w_v(b_{R_v})/2$$

and

$$r_{R_v} = \mu^x_{vehicle} + w_v(b_{R_v})/2, \tag{10}$$

where $\mu^x_{vehicle}$ is the x position of $L_{vehicle}$.

4 License Plate Detection

Once the ROI of a vehicle can be located, the SVD system needs to check whether a license plate exists within this ROI. This paper uses an edge-based method to detect the regions having a higher edge density as license plate candidates.

For a pixel $p(x,y)$, its edge response can be calculated by using a Sobel operator only in the x-direction as follows:

$$I_x(x,y) = \frac{|I(x+1,y) - I(x-1,y)|}{2}. \tag{11}$$

Then, we can get an edge map of $e(x,y)$ where $e(x,y)$ is one if $p(x,y)$ is an edge point and zero, otherwise. To calculate the edge density of an area, we take advantages of integral image to scan and find license candidates extreme quickly from $I(x,y)$. Next a predefined threshold T_{edge} is used to exclude impossible license plate candidates whose edge densities are lower than the threshold. If the length-to-width ratio of the remained region does not fit a range, i.e., [1/5, 9/10], it will be further filtered out. However, the size of a license plate R in $I(x,y)$ is unknown. Thus, a smaller mask M with the dimension $m \times m$ is used to scan all pixels in $I(x,y)$. Each M will be kept if its edge density is higher than T_{edge}. When scanning in the horizontal direction, the shift jump between two adjacent masks is $4m/5$ for efficiency improvement. Another shift jump in the y direction is $m/2$. Two masks whose edge densities are higher than T_{edge} are merged together if they are adjacent in the x or y

direction. After merging, a set of possible license plate candidates R_i with different dimensions can be extracted. Next, R_i is excluded if its geometrical properties do not meet a vehicle plate specification. The specification demands the merged candidate should have a predetermined length-to-width ratio and be large enough. Then, a great quantity of unqualified candidate plate areas can be filtered out.

There are still some false license plate regions to be extracted with the above filtering rules. Then, we use our previous work [21], [24] to train an Adaboost-based classifier to quickly verify each remained license plate candidate. Traditional Adaboost-based methods should scan all pixels across all scales to find all possible object candidates. In our scheme, this classifier is only applied into the verification stage. Impossible candidates have been almost filtered out in our scheme before verification. Thus, each desired license plate can be found extremely fast.

5 Experimental Results

To evaluate the performances of our proposed system, an automatic system for detecting suspected vehicles was implemented in this paper. The first set of experiments was performed to evaluate the performance of vehicle detection using our symmelet-based scheme. The proposed method does not require any motion features and thus can work well to detect vehicles from only single image. Figure 5 shows the results of vehicle detection from front views. The ROI of each vehicle is shown by a red rectangle. Another challenging problem is to detect vehicles from non-front views. Figure 6 shows the case of vehicle detection when side views were handled. When the viewing angle ranges from -25^0 to 25^0, our method still works.

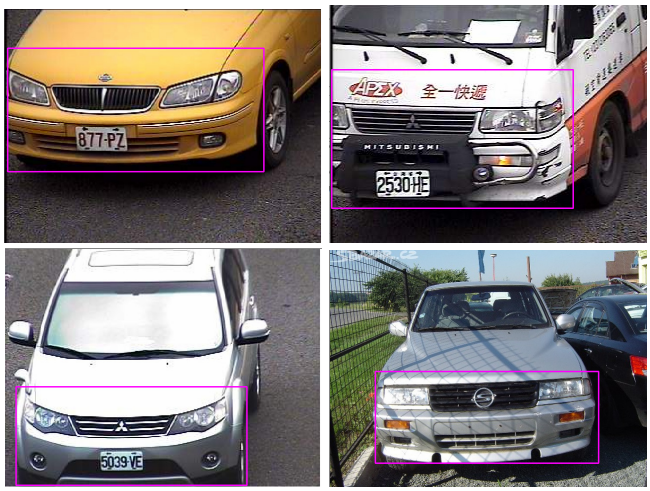


Figure 5. Results of vehicle detection when front views were handled

Figure 6. Results of vehicle detection when side views were handled

Weather is another important factor to affect the performance of vehicle detection. When the sun is strong, various tree or building shadows will be generated and casted on the analyzed vehicles. The existence of shadow will disturb the accuracy of vehicle detection. Figure 7 shows the detection results when various shadows were casted on the analyzed vehicles. Rainy day is another challenge because the rain will blur the strength of a feature and thus lead to the failure of vehicle detection. Figure 8 shows the cases of vehicle detection under rainy days. Table 1 shows the accuracy comparisons of vehicle detection under different weathers. The best performance is got from the cloudy days.

Figure 7. Results of vehicle detection when shadows were casted on the analyzed vehicles

Figure 8. Results of vehicle detection from rainy scenes

Table 1. Accuracy analyses of vehicle detection under different weathers.

Conditions	Sunny	Cloudy	Rainy
Correct	5075	3152	2055
Total	5230	3273	2185
Accuracy	97.03%	96.30%	94.13%

In addition to weather, night lighting is another factor to affect the performance of vehicle detection. Figure 9 shows the cases of vehicle detection at night time. Table 2 shows the accuracy comparisons of vehicle detection at day time and night time. Clearly, the best performance is got from the scenes captured at daytime.

Figure 9. Results of vehicle detection at night time

Table 2. Accuracy analyses of vehicle detection at daytime and nighttime

Conditions	Day Scenes	Night Scenes
Correct	7567	3175
Total	7345	2,984
Accuracy	97.07%	93.98%

As to the performance evaluation about our SVD system, there are 30,795 images collected for vehicles with license plates, and 31,275 images for vehicles without license plates. All the testing images were obtained from Taiwan's FETC company [1]. Table 3 lists the numbers of vehicles used for performance evaluation of our SVD system with/without license plates at daytime or nighttime. Figure 10 shows the detection results of our SVD system to detect vehicles with license plates at daytime. Due to the limit of paper size, only few testing results are shown in Figure 10. Figure 11 shows the detection results of our SVD system to detect vehicles with plates at nighttime. Figure 12 shows two failure cases of our SVD system to detect moving violations due to plate occlusions. Table 4 shows the accuracy analyses of our SVD system to detect vehicles with plates at daytime or nighttime. Clearly, our SVD system performs better at daytime than nighttime due to well lighting conditions.

Table 3. Numbers of vehicles used for testing with/without plates at daytime or nighttime

Conditions	Daytime	Nighttime	Total
With Plates	18955	11840	30,795
No Plate	16967	14308	31,275
Total	35922	26148	62070

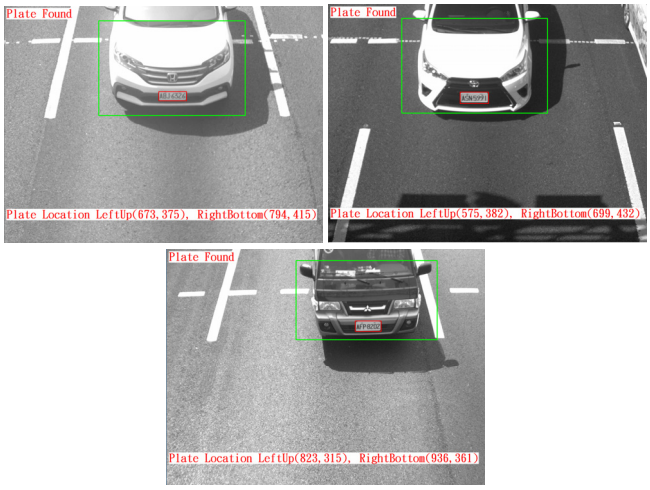


Figure 10. Detection results of our SVD system to detect vehicles moving with plates at daytime

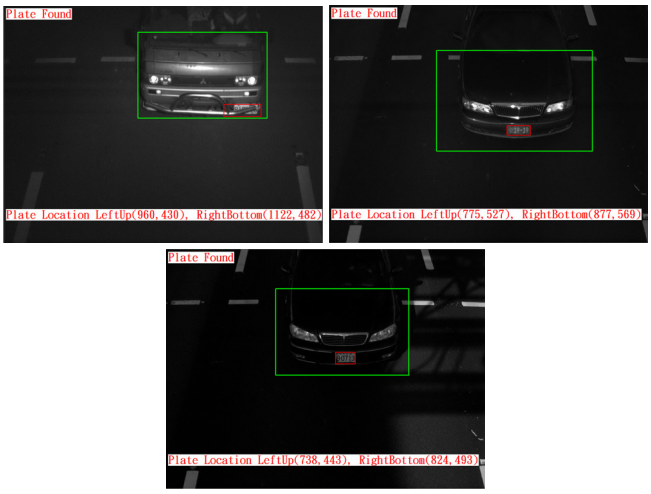


Figure 11. Detection results of our SVD system to detect vehicles moving with plates at nighttime

Figure 12. Failure cases of our SVD system to detect moving violations

Table 4. Accuracy analyses of our SVD system to detect vehicles with plates at daytime or nighttime

Conditions	With Plate	No Plate	Total	Accuracy
Daytime	18189	766	18955	95.96%
Nighttime	11197	643	11840	94.57%

Figure 13 shows the detection results of our SVD system to detect moving violations when vehicles moved without plates at daytime. Figure 14 shows another set of results of our SVD system to detect moving violations at nighttime. Table 5 shows the accuracy analyses of our SVD system to detect moving violations when vehicles moved without any plate. Similar to Table 4, our SVD system performs better at daytime than nighttime due to well lighting conditions. Experimental results have proved the superiority of our proposed system in suspected vehicle detection for vehicle moving without license plates.

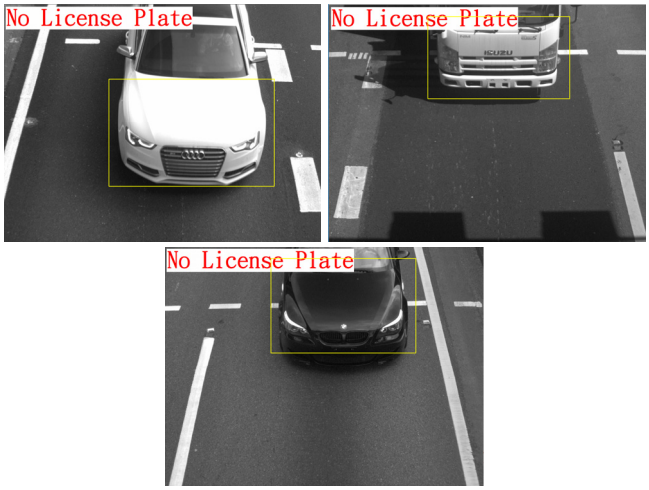


Figure 13. Detection results of our SVD system to detect vehicles moving without license plate at daytime

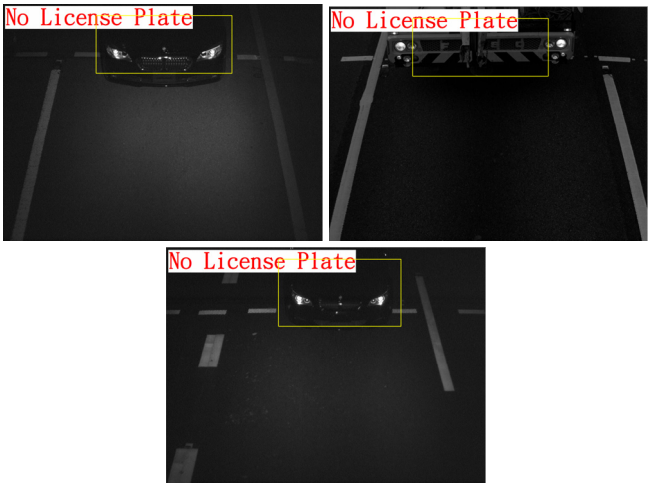


Figure 14. Detection results of our SVD system to detect vehicles moving without hanging license plates at nighttime

Table 5. Accuracy analyses of our SVD system to detect moving violations for moving without plates

Conditions	Plate	No Plate	Total	Accuracy
Daytime	15587	1380	16967	91.87%
Nighttime	12397	1911	14308	86.64%

6 Conclusion

This paper has represented a novel symmelet-based SVD system to detect vehicles travelling on roads without a license plate. We have tested our method on a large dataset with real practical images and have proved the value, efficiency, and effectiveness of our proposed system in detecting suspected vehicles without hanging license plates travelling on roads.

References

[1] FETC, http://www.fetc.net.tw/en/.

[2] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359, June, 2008.

[3] A. Faro, D. Giordano, C. Spampinato, Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, pp. 1398-1412, December, 2011.

[4] H. Unno, K. Ojima, K. Hayashibe, H. Saji, Vehicle Motion Tracking Using Symmetry of Vehicle and Background Subtraction, *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007, pp. 1127-1131.

[5] A. Jazayeri, H.-Y Cai, J.-Y. Zheng, M. Tuceryan, Vehicle detection and tracking in car video based on motion model, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 2, pp. 583-595, June, 2011.

[6] G. L. Foresti, V. Murino, C. Regazzoni, Vehicle recognition and tracking from road image sequences, *IEEE Transactions on Vehicular Technology*, Vol. 48, No. 1, pp. 301-318, January, 1999.

[7] L.-W. Tsai, J.-W. Hsieh, K.-C. Fan, Vehicle Detection Using Normalized Color and Edge Map, *IEEE Transactions on Image Processing*, Vol. 16, No. 3, pp. 850-864, March, 2007.

[8] W.-C. Chang, C.-W. Cho, Online boosting for vehicle detection, *IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics*, Vol. 40, No. 3, pp. 892-902, June, 2010.

[9] S. Sivaraman, M. Trivedi, A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, pp. 267-276, June, 2010.

[10] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, September, 2014.

[11] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 2014, pp. 580-587.

[12] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot multibox detector, In *European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21-37.

[14] M. Shridhar, J. W. V. Miller, G. Houle, L. Bijnagte, Recognition of license plate images: issues and perspectives, *Proc. of the Fifth International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp. 17-20.

[15] K. K. Kim, K. I. Kim, J. B. Kim, H. J. Kim, Learning-based approach for license plate recognition, *Proc. of IEEE Workshop on Neural Networks for Signal Processing*, Vol. 2, Sydney, NSW, Australia, 2000, pp. 614-623.

[16] H. A. Hegt, R. J. Haye, N. A. Khan, A high performance license plate recognition system, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, USA, 1998, pp. 4357-4362.

[17] M. Yu, Y. D. Kim, An approach to Korean license plate recognition based on vertical edge matching, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN, USA, pp. 2975-2980, 2000.

[18] D. S. Kim, S. I. Chien, Automatic car license plate extraction using modified generalized symmetry transform and image warping, *Proc. of IEEE International Symposium on Industrial Electronics*, Vol. 3, Pusan, Korea, 2001, pp. 2022-2027.

[19] S. Du, M. Ibrahim, M. Shehata, W. Badawy, Automatic license plate recognition: A state-of-the-art review, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 2, pp. 311-325, February, 2013.

[20] Y. Dai, H. Q. Ma, J. Liu, L. Li, A high performance license plate recognition system based on the web technique, *Proc. of IEEE Proceedings Intelligent Transportation Systems*, Oakland, CA, USA, 2001, pp. 325-329.

[21] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, Morphology-based license plate detection from complex scenes, in *Proc. IEEE 16th International Conference on Pattern Recognition*, 2002, Vol. 3, Quebec City, QC, Canada, 2002, pp. 176-179.

[22] C. Gou, K.-F. Wang, Y.-J. Yao, Z. Li, Vehicle License Plate Recognition Based on Extremal Regions and Restricted Boltzmann Machines, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 4, pp. 1096-1107, April, 2016.

[23] H. Li, C. Shen, Reading car license plates using deep convolutional neural networks and LSTMs, In: arXiv: 1601.05610, January, 2016.

[24] C.-C. Chen, J. W. Hsieh, License plate recognition from low-quality videos, *Proc. of IAPR Conference on Machine Vision Applications*, Tokyo, Japan, 2007, pp. 122-125.

[25] J.-W. Hsieh, L.-C. Chen, D.-Y. Chen, Symmetrical SURF and its Applications to Vehicle Detection and Vehicle Make and Model Recognition, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, pp. 6-20, February, 2014.

Biographies

Jun-Wei Hsieh is currently a professor at the College of AI in National Chiao-Tung University (2019.08.01). He was a professor and dean of the department of Computer Engineering in National Taiwan Ocean University (2009.8.01~present). His research fields include AI, Deep learning, image and video processing, machine learning, computer vision, etc.

Hung Chun Chen graduated from NTOU with master degree of Computer science and study Computer Vision. Now he is working at AIC in Taoyuan.

Ping-Yang Chen is studying for his Ph.D. degree in the Department of computer science at National Chiao Tung University from 2019 to the present. In July 2019, he received his master's degree in computer science and engineer at National Taiwan Ocean University. He has worked in the fields of image processing, computer vision, pattern recognition, and deep learning for more than five years. In 2020, he got the best paper awards from NCWIA 2020 and CVGIP 2020, respectively.

Shiao-Peng Huang received his M.S. degree in computer science and information engineering from National Central University in 2008. He is currently a researcher at Chunghwa Telecom Laboratories in Taiwan. His research interests include machine learning, and computer vision.