

Vehicle Color Classification Under Different Lighting Conditions Through Color Correction

Jun-Wei Hsieh, *Member, IEEE*, Li-Chih Chen, Sin-Yu Chen, Duan-Yu Chen,
Salah Alghyline, and Hui-Fen Chiang

Abstract—This paper presents a novel vehicle color classification technique for classifying vehicles into seven categories under different lighting conditions via color correction. First, to reduce lighting effects, a mapping function is built to minimize the color distortions between frames. In addition to color distortions, the effect of specular highlights can also make the window of a vehicle appear white and degrade the accuracy of vehicle classification. To reduce this effect, a window-removal task is performed to make vehicle pixels with the same color more concentrated on the analyzed vehicle. Thus, a vehicle can be more accurately classified into its corresponding category even when it is shone by strong sunlight. One major problem in vehicle color classification is that there are many shade colors; for example, white versus silver and black versus navy. Traditional methods lack the ability to classify vehicles with shade colors because a wrong classifier is designed by putting vehicles with the same label together even though their chromatic attributes are different. To treat this problem, a novel tree-based classifier is designed for classifying vehicles into chromatic/nonchromatic classes with their nonchromatic strengths and then into detailed color classes with their color features. The separation can significantly improve the accuracy of vehicle color classification even that vehicles are with various shade colors and captured under different lighting conditions.

Index Terms—Vehicle color classification, color correction, SVM, vehicle window removal.

I. INTRODUCTION

VIDEO surveillance in public spaces has attracted immense attention in recent years because of its promising capabilities for crime prevention and security. For example, in the VSAM (Video Surveillance And Monitoring) project [1], a wide range of advanced surveillance techniques were developed for real-time moving object detection, tracking, and counting, recognition of generic object classes, human activity recognition, and so on. To overcome the limited

Manuscript received April 11, 2014; revised July 24, 2014; accepted August 26, 2014. Date of publication September 16, 2014; date of current version November 26, 2014. This work was supported in part by the National Science Council of Taiwan under Grant NSC-97-2221-E-155-060 and in part by the Ministry of Economic Affairs under Contract 98-EC-17-A-02-s1-032 and MOEA-102-E0616. The associate editor coordinating the review of this paper and approving it for publication was Prof. Alexander Fish.

J.-W. Hsieh, S. Alghyline, and H.-F. Chiang are with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan (e-mail: shieh@ntou.edu.tw; salahshaman2007@gmail.com; chelly.chiang@gmail.com).

L.-C. Chen is with the Department of Electrical Engineering, Lee-Ming Institute of Technology, Taipei 243, Taiwan (e-mail: lcchen@mail.lit.edu.tw).

S.-Y. Chen and D.-Y. Chen are with the Department of Electrical Engineering, Yuan Ze University, Zhongli 320, Taiwan (e-mail: polar1010@gmail.com; dychen@saturn.yzu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSEN.2014.2358079

field of view of a camera, many cameras should be used for monitoring a wide-area scene. However, because of differences in lighting and camera distortions, the visual properties of an object will differ significantly for different cameras. Thus, tracking and identifying a target moving across cameras is very challenging [2]. It is especially important for police to be able to confidently track a vehicle or search escape cars across different cameras.

To treat the problem, different space-time cues are used to build feature correspondences and then obtain different spatial relations between cameras for object tracking. For example in [3], Chen *et al.* proposed a batch learning algorithm for determining the topology of camera networks; then, they extracted the appearance relationships using brightness transfer functions to track targets among multiple disjoint cameras. In [4], Pflugfelder and Micusik mapped the tracking problem to a tree structure and then used a branch-and-bound scheme to associate the trajectories of an object and track its positions across cameras. In addition, Lian, Lai, and Zheng [5] proposed an online correspondence updating method for building spatial relations and then tracking pedestrians across non-overlapping cameras. In [6], Lian *et al.* integrated color intensity (CI) and a distance-based local binary pattern (DLBP) to construct a novel CI_DLBP descriptor for matching and tracking pedestrians across disjoint camera views. In real-world applications, although the color of an object changes across different cameras, if a proper color transform can be found for correcting this visual distortion, the color can be a very useful cue for object identification. In addition, when a crime or accident happens, color is often the only feature to be remembered by the witness and provided for the policemen to search escape cars.

Maintaining color constancy between cameras is another important technique for solving the problem of matching objects using their colors among different cameras. Finlayson, Drew, and Funt [9] proposed a sensor transformation called spectral sharpening for constructing a set of sensor-sensitive functions to maintain color constancy between cameras. Buchsbaum [10] proposed a “Gray-World” model for obtaining some invariant surface color descriptors by using a scene-averaging technique. Weijer and Gevers [11] modified this model and then proposed an edge-based model, called “Gray-Edge”, for measuring the color constancy between objects. Gehler *et al.* [12] used a Bayesian approach to improve the Gray-World algorithm such that the true reflectance of visible surfaces in an image can be

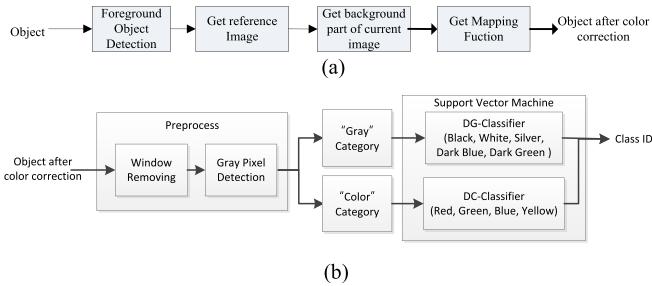


Fig. 1. Overview of the proposed system for vehicle classification. (a) Color correction. (b) Vehicle classification.

more accurately estimated. Xiang *et al.* [13] proposed a model-based transformation between two images for reducing the color distortions between two cameras. Matsushita [14] constructed an illumination eigenspace for analyzing possible lighting variations and eliminating unwanted shadow effects. For object tracking, Madden, Cheng, and Piccardi [7] proposed a histogram-stretching technique for minimizing the color differences between two cameras such that the same object can be well tracked across different cameras. This scheme works well when the differences between object colors are large. However, it will fail in vehicle color classification because there are many vehicles with shade colors. The colors of these vehicles will often be over-stretched by this method.

There are many challenges in the task of vehicle color classification because there are various changes in lighting, shadow, the time of vehicle capturing, and camera setting such automatic white balance. Moreover, the vehicle colors are not uniformly distributed and may not actually represent the vehicle color. In addition, strong light reflections on a vehicle often result in a fake “white” color. More importantly, there are many shade (or confused) colors in vehicle color classification such as dark *vs.* dark navy, white *vs.* silver, dark *vs.* dark green, and so on. In the literature, different methods [23]–[27] have been proposed to tackle the above challenges. For example, Brown [23] investigated several color features from different color spaces to evaluate their effectiveness in vehicle color classification. Furthermore, Fang *et al.* [24] extracted features from the HSI color space and then used back propagation networks to classify vehicles. In [25], Wang *et al.* detected the taillights of a vehicle and then extracted its color features from the CIELab color space to classify vehicles via a KNN classifier. Yang *et al.* [26] extracted features from the HSV and RGB spaces to classify colorful vehicles via a rule-based classifier. In [27], Baek *et al.* proposed a brute-force scheme to divide the hue and saturation values of the HSV color space into different bins from which a classifier was trained for vehicle color classification. One problem in the above designed classifiers is the lack of ability to deal with shade colors in vehicle color classification. The above methods tend to train a classifier wrongly by putting vehicles with the same label together even though their chromatic attributes are different. Then, mistakes often happen when vehicles are with less chromatic colors, e.g., “dark navy” but need to be classified to a chromatic color category, e.g., “blue”.

This paper presents a novel color-correction technique to correct vehicles’ colors in real time and then classify each

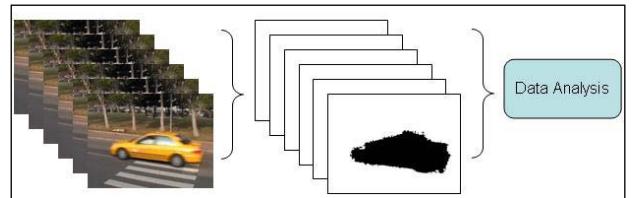


Fig. 2. A series of input frames was collected for background modeling and vehicle extraction.

input vehicle into its corresponding color category. Fig. 1 shows the flowchart of our proposed system. First, a novel color-correction scheme for minimizing the color distortion of input frames is proposed. Then, a novel vehicle classification scheme for classifying vehicles into different color categories is proposed. Details of the two methods are shown in Fig. 1(a) and (b). The color-correction technique (see (a)) builds a mapping function to minimize the color distortions between frames. In real cases, strong light reflections on a vehicle often result in a fake “white” color. Thus, after color correction, at the stage of vehicle classification (see (b)), a window-removal method is applied to make vehicle pixels with the same color more concentrated on the foreground region. To well address the problem of vehicles with shade colors, this paper proposes a new concept that vehicles with different chromatic attributes should be separately trained even though they belong to the same color category. It means the vehicles with lower chromatic strengths should be first separated from the one with higher chromatic strengths. After that, each group with different chromatic strengths is then classified into detailed labels by its corresponding finer classifier. Thus, a novel tree-based classifier is used to solve this confusion problem, *i.e.*, vehicles with shade colors. The proposed scheme can significantly reduce the effects of color distortion. Thus, even under different lighting conditions, each vehicle can still be correctly classified into its color category. The classification result can be used for suspicious-vehicle detection if a vehicle’s color is not the color associated with its license plate number. In addition, when the camera spatial relations are known in advance, a suspicious vehicle can be forward- or backward-tracked using its color for security maintenance and vehicle counting. Experimental results demonstrate the feasibility and superiority of the proposed approach in vehicle classification under different cameras and lighting conditions.

The remainder of the paper is organized as follows. In the next section, the details of the color correction are described. Then, Section III describes the details of the feature extraction and vehicle classification using SVMs. Section IV reports the experimental results. Finally, some conclusions are presented in Section V.

II. COLOR CORRECTION

The apparent color of a vehicle changes as a function of time, space, and lighting conditions. Thus, in the material that follows, a novel color correction scheme is proposed for minimizing the effects of lighting changes.

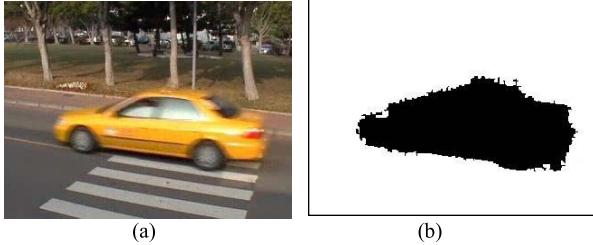


Fig. 3. Result of foreground extraction. (a) Input frame. (b) Result of background subtraction.



Fig. 4. A vehicle is with different colors under various lighting conditions.

A. Foreground Vehicle Extraction

Before color correction, each foreground vehicle is extracted from its background. In Fig. 2, a series of frames is collected for constructing the background. Here, this paper uses the codebook technique [15] to perform the background modeling and subtraction tasks. The vehicle can be also detected from single image without background subtraction by using our previous work [29]. Fig. 3 shows the result of the background subtraction. (b) is the result of the background subtraction performed on (a). After several morphological operations and a connected-component analysis, only the remaining foreground region with the largest area is used for vehicle analysis.

B. Color Mapping and Correction

For a vehicle, its colors will appear different under different lighting conditions and viewpoints. Fig. 4 shows an example that the colors of a vehicle are different under various lighting conditions. Before vehicle classification, a novel color correction scheme is needed to correct for lighting changes.

Reinhard *et al.* [16] presented a simple but effective image inpainting technique that transfers color characteristics from a source I_s to a target image I_t for color correction. Let μ_s and μ_t be the means of I_s and I_t , respectively. To reduce the color difference between I_s and I_t , these authors proposed a mapping function $g(p)$ for transferring the color of a pixel p at I_t in the *LAB* color space to I_s :

$$g(p) = \mu_s + \frac{\sigma_s}{\sigma_t}(I_t(p) - \mu_t), \quad (1)$$

where σ_s and σ_t are the variances of I_s and I_t , respectively. Eq. (1) is a global color-mapping function because the color statistics are calculated by using all pixels in the analyzed image. When images contain different color regions, the mapping function cannot distinguish different color statistics among regions and will cause unnatural and saturated results. To better correct images, Tai, Jia, and Tan [17] defined several local transfer functions and estimated their parameters via an EM algorithm. The iterative updating process is quite time-consuming and not suitable for real-time applications. In [18],

an optimal transfer function is constructed by integrating different local transfer functions built from many models and regions. Because many models are involved, this correction method is also time-consuming. To fit the real-time requirement, we reformulate this color correction method to improve its efficiency and effectiveness.

After rewriting Eq.(1), we have

$$g(p) = \frac{\sigma_s}{\sigma_t} I_t(p) + (\mu_s - \frac{\sigma_s}{\sigma_t} \mu_t) = \alpha I_t(p) + \beta, \quad (2)$$

where $\alpha = \frac{\sigma_s}{\sigma_t}$ and $\beta = \mu_s - \frac{\sigma_s}{\sigma_t} \mu_t$. Thus, $g(I_t)$ becomes a linear model parameterized by α and β to approximate I_s . Then, we can obtain the parameters α and β by minimizing the following energy function:

$$(\alpha, \beta) = \arg \min_{\alpha, \beta} \sum_p [I_s(p) - \alpha I_t(p) - \beta]^2. \quad (3)$$

By setting the first derivatives of the right term of Eq.(3) (to α and β) to zero, we obtain

$$\begin{cases} \sum I_s I_t - \alpha \sum I_t^2 - \beta \sum I_t = 0, \\ \sum I_s - \alpha \sum I_t - N\beta = 0, \end{cases} \quad (4)$$

where N is the size of I_s . After solving Eq.(4), α and β are obtained as follows:

$$\alpha = \frac{N\mu_s \mu_t - \sum I_t I_s}{N\mu_t^2 - \sum I_t^2} \quad \text{and} \quad \beta = \frac{\mu_t(\sum I_t I_s) - \mu_s \sum I_t^2}{N\mu_t^2 - \sum I_t^2}. \quad (5)$$

Similar to Eq. (2), Eq. (5) is also a global color mapping between I_t and I_s and works based on the assumption that a constant correction function exists between the two images. However, in complex scenes, this assumption does not always hold because of different scene content and image-capturing devices. To solve this problem, we segment the source image I_s to several regions R_i^s using the algorithm proposed by Felzenszwalb and Huttenlocher [19]. Then for each region R_i^s , its corresponding region R_i^t on I_t is obtained using one-to-one pixel mapping. A one-to-one mapping means that a pixel in R_i^s and its correspondence in R_i^t share the same coordinate. Thus, R_i^s and R_i^t have the same size and their pixel correspondence is one-to-one. Then, the local color mapping function g_i from R_i^t to R_i^s can be written as

$$g_i(p) = \alpha_i I_t(p) + \beta_i \quad \text{for all } p \in R_i^t. \quad (6)$$

Let μ_i^t and μ_i^s be the color means of R_i^t and R_i^s , respectively. Similar to (5), the parameters α_i and β_i to parameterize g_i can be obtained as

$$\alpha_i = \frac{|R_i| \mu_i^s \mu_i^t - \sum_{p \in R_i^t} I_t(p) I_s(p)}{|R_i| (\mu_i^t)^2 - \sum_{p \in R_i^t} I_t^2(p)}$$

and

$$\beta_i = \frac{\mu_i^t [\sum_{p \in R_i^t} I_s(p) I_t(p)] - \mu_i^s \sum_{p \in R_i^t} I_t^2(p)}{|R_i| (\mu_i^t)^2 - \sum_{p \in R_i^t} I_t^2(p)}, \quad (7)$$

where $|R_i|$ denote the size of R_i . By considering the contribution of each color transfer function g_i , a weighted color

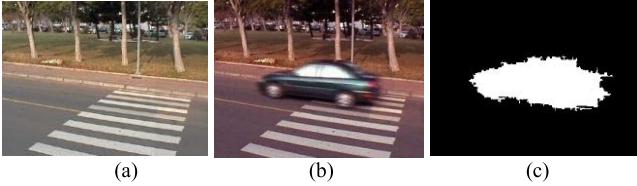


Fig. 5. Color Correction. (a) Reference image. (b) Image with color distortions. (c) Pixels with black color showing the background part of (b).

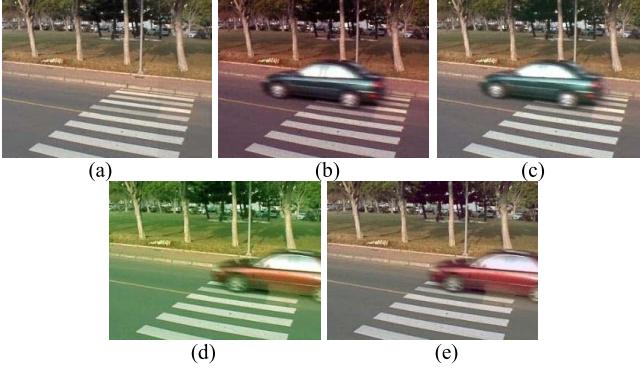


Fig. 6. Results of color correction. (a) Source image. (b) and (d): Input frames. (c) and (e): Results of color correction from (b) and (d), respectively.

correction function $G(p)$ can be formed. The importance of g_i for a pixel p is calculated according to the color distance between p and the mean color μ_i^s of R_i^s . Let $\|x - y\|$ denote the Euclidean distance between two colors x and y . Then, the importance of g_i for p is defined by

$$w_i(p) = \exp(-\|g_i(p) - \mu_i^s\|^2). \quad (8)$$

Assume there are N_R segmented regions. Thus, there are N_R local mapping functions. In addition, let $\bar{w}_i(p)$ denote the normalized version of $w_i(p)$,

$$\bar{w}_i(p) = \frac{w_i(p)}{\sum_{i=1}^{N_R} w_i(p)}.$$

Then, the weighted correction function $G(p)$ is defined as

$$G(p) = \sum_{i=1}^{N_g} \bar{w}_i(p) g_i(p). \quad (9)$$

Using Eq. (9), we can map each input frame I_t to I_s for color correction. However, Eq. (9) works poorly for color correction if I_t contains some foreground objects. Thus, before color correction, all foreground objects must be removed. In Fig. 5(c), the vehicle region is extracted and removed using background subtraction. Only the pixels of the background regions (which are colored black) in I_t are used for calculating g_i and then building $G(p)$ for color correction.

In real cases, the background (denoted by I_b) must be adaptively updated for foreground object detection because the background changes with time. The source image I_s is fixed during the period of color correction. Thus, I_b is not equal to I_s and is just used for background subtraction rather than



Fig. 7. Vehicle windows with black color.



Fig. 8. Vehicle window colors became white due to the effect of specular highlights.

color correction. Fig. 6 shows the results of color correction. (a) is the source image. (c) and (e) are the results of the color correction performed on (b) and (d), respectively. After color correction, the color characteristics of (c) and (e) are similar to those of (a). For a vehicle moving across different cameras, the proposed correction method will not correct its colors from one camera to another camera. Because we just need to ensure its color label to be correctly classified under the same camera, the source image I_s is differently chosen from each camera but fixed under the same camera.

III. VEHICLE CLASSIFICATION

After color correction, different classifiers are trained for classifying vehicles into different categories. In Fig. 7, the vehicle windows appear black. The black color will cause the vehicles to be misclassified as “black”. However, the effect of specular highlights also often makes a vehicle’s windows appear a fake white color. In Fig. 8, this effect makes the vehicle windows appear white and leads to the misclassification of the vehicles as “white”. To reduce this effect, the vehicle windows must first be removed. After their removal, different vehicle classifiers can then be trained to classify vehicles into different color categories.

A. Vehicle Window Removal

Let R denote the vehicle extracted using a background subtraction technique. To remove the vehicle’s windows, the orientation θ_R of its major axis must first be detected. This paper uses a moment-based approach to detect θ_R . First, the central moment of R is defined as

$$(\mu_{p,q})_R = \sum_{(x,y) \in R} (x - \bar{x})^p (y - \bar{y})^q,$$

where $(\bar{x}, \bar{y}) = (\frac{1}{|R|} \sum_{(x,y) \in R} x, \frac{1}{|R|} \sum_{(x,y) \in R} y)$ and $|R|$ is the area of R . Then, as shown in Fig. 9, the orientation θ_R of R can be obtained using the equation

$$\theta_R = \arg \min_{\theta} \sum_{(x,y) \in R} [(x - \bar{x}) \sin \theta - (y - \bar{y}) \cos \theta]^2. \quad (10)$$

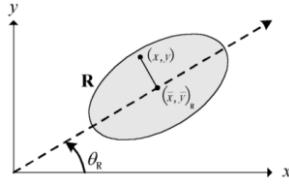
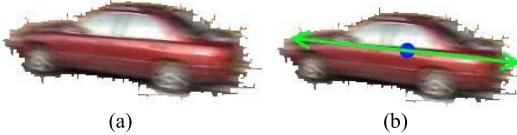
Fig. 9. The gravity center $(\bar{x}, \bar{y})_R$ and orientation θ_R of a vehicle R .

Fig. 10. (a) Input vehicle. (b) Cutting line used for removing the vehicle window.



Fig. 11. Result of vehicle window removal.

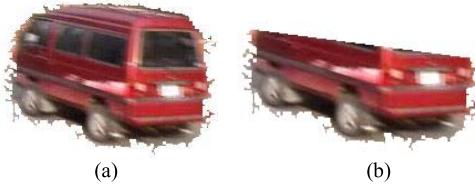


Fig. 12. Result of vehicle window removal from a minivan. (a) Input image. (b) Result of window removal.

By setting the first derivative of Eq.(10) to zero, the direction θ_R can be estimated by

$$\theta_R = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right]. \quad (11)$$

If θ_R ranges from $-\pi/4$ to $\pi/4$, R is considered “horizontal”; otherwise, its orientation is “vertical”. Actually, the hood region of a vehicle will include most of color features. If the orientation of R is “vertical”, its moving direction should be also known for determining which part of R is the real hood region. The moving direction can be easily obtained by checking the center moving of R across two adjacent frames via a simple tracking scheme.

Let L_{cut} denote the cutting line used to remove the windows of R . If R is “horizontal”, L_{cut} is the line with the orientation θ_R that passes through the center (\bar{x}, \bar{y}) . In Fig. 10, (a) is the input vehicle and (b) shows the line L_{cut} that passes through the center (\bar{x}, \bar{y}) . For each pixel p in R , if p is below L_{cut} , p is added to the remaining vehicle \bar{R} ; otherwise, it is eliminated. Fig. 11 shows the result of vehicle window removal performed on the image shown in Fig. 10. Fig. 12 shows the result of window removal for a minivan. If R is “vertical”, L_{cut} is the line with orientation $(\theta_R + \pi/2)$ that passes through the center (\bar{x}, \bar{y}) . L_{cut} separates R into an upper part ‘ Ω ’ and a lower part ‘ \mathcal{U} ’. Because R moves along a road, its direction of movement can be determined using the

Algorithm 1 Algorithm for Vehicle Window Removal

Input: a vehicle R with a moving direction \vec{d} .
Output: the remaining vehicle \bar{R} after window removal.
Step1: Estimate θ_R using Eq. (11). If θ_R ranges from $-\pi/4$ to $\pi/4$, go to Step 2; otherwise, go to Step 3.
Step2:
2.1: Find L_{cut} with θ_R and the center (\bar{x}, \bar{y}) .
2.2: Add each pixel p in R to the set \bar{R} if the pixel is below L_{cut} .
2.3: Return \bar{R} .
Step3:
3.1: Find L_{cut} with the orientation $(\theta_R^+ \pi/2)$ and the center (\bar{x}, \bar{y}) .
3.2: Separate R into an upper part Ω and a lower part \mathcal{U} .
3.3: If the direction \vec{d} of movement of R is toward Ω , $\bar{R} = \Omega$;
else $\bar{R} = \mathcal{U}$.
3.4: Return \bar{R}

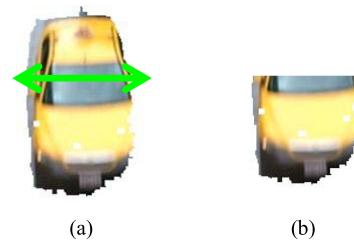


Fig. 13. Result of vehicle window removal when the orientation is ‘vertical’. (a) Input Frame. (b) Result of the proposed algorithm.



Fig. 14. Seven color appearance categories of vehicle used for color classification.

difference in its position between the current frame and the previous frame. If the direction of movement of R is toward Ω , Ω will be retained for further color classification; otherwise, \mathcal{U} is selected. In Fig. 13, (a) is the input vehicle with a “vertical” orientation. Because the vehicle moves down, \mathcal{U} is retained as the remaining vehicle \bar{R} . (b) is the result of the window removal. A tree-based classifier will be trained later for classifying \bar{R} into different categories. In Algorithm 1, details of the window removal algorithm are described.

B. Gray Pixel Identification

After window removal, the following stage is to classify the remaining vehicle \bar{R} into different categories. In this paper, seven color categories (“blue”, “green”, “red”, “yellow”, “white”, “silver”, “black”) are collected for vehicle classification (shown in Fig. 14). In this set, there are many shade colors to be easily misclassified. For example, a “navy” vehicle in the “blue” category is often misclassified into “black”. Traditional methods [23]–[27] lack the ability to deal with this problem, *i.e.*, vehicles with shade colors, because a wrong classifier is designed by putting vehicles with the same label together even though their chromatic attributes are different. To well address

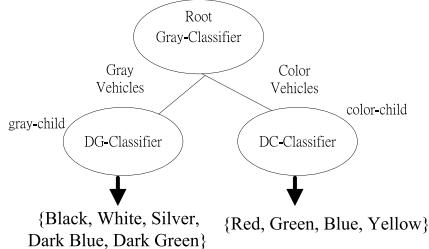


Fig. 15. A tree structure for vehicle classification.

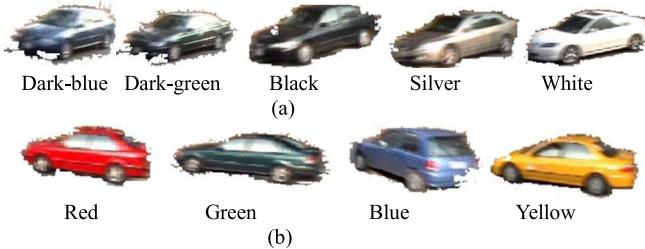


Fig. 16. Vehicles classified to “gray” and “color” classes. (a) Gray class. (b) Color class.

the problem, this paper proposes a new concept that vehicles with different chromatic attributes should be separately trained. It means the vehicles with lower chromatic strengths, *e.g.*, “navy”, should be first separated from the one with higher chromatic strengths, *e.g.*, “blue”. After the separation, elements in a specific subgroup are then classified into detailed labels by using its corresponding finer classifier. A novel tree-based classifier is then inspired and built to tackle this confusion problem in color classification.

Fig. 15 shows the structure of our proposed tree-based vehicle classifier. The classifier splits the “blue” and “green” categories into four categories, “blue”, “dark blue”, “green”, and “dark green”. Then, at the root node, a G-classifier (G for “Gray”) is designed to classify vehicles according to their non-chromatic strengths into the “gray” and “color”subgroups. As shown in Fig. 16, the “gray” subgroup contains the “black”, “silver”, “white”, “dark blue”, and “dark green” categories, respectively. The “color” subgroup contains the “red”, “green”, “blue”, and “yellow” categories, respectively. For the child nodes in Fig. 15, the “DG-classifier” and “DC-classifier” are then trained to classify elements in the “gray” and “color”subgroups, respectively. Here, the DG-classifier and DC-classifier denote the “detailed gray classifier” and “detailed color classifier”, respectively. The details of the G-classifier will be described in Section III.C. The frameworks of the DG-classifier and DC-classifier are discussed in Section III.D.

C. Non-Chromatic Pixel Identification

To define the G-classifier, we first normalize a pixel p as follows:

$I_R = r/(r+g+b)$, $I_G = g/(r+g+b)$, and $I_B = b/(r+g+b)$, where $(r, g, b)^t$ is the color vector of p . If $(r, g, b)^t$ is a gray color, it should be very close to the gray axis, $(1/3, 1/3, 1/3)^t$.

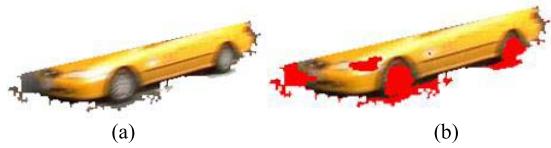


Fig. 17. (a) Input vehicle. (b) Result of gray pixel detection.

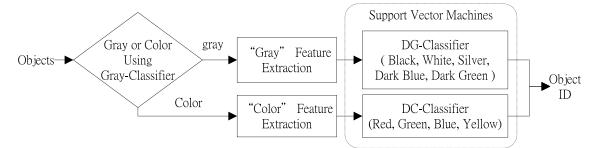


Fig. 18. Two classifiers were trained for classifying vehicles to different color appearance categories. The DG-classifier is used for detailing the “gray” subgroup and the DC-classifier is used for detailing the “color” subgroup.

Thus, the distance between p and the gray axis is defined as

$$d_p = (I_R - 0.333)^2 + (I_G - 0.333)^2 + (I_B - 0.333)^2.$$

In addition, if p belongs to a gray pixel, its color channels should all have values similar to its mean u_p , where $u_p = (r+g+b)/3$. Then, the distance between p and u_p is defined as follows:

$$d_{mean} = (r - u_p)^2 + (g - u_p)^2 + (b - u_p)^2. \quad (12)$$

Let u_{mean} and u_d denote the means of d_{mean} and d_p , respectively (calculate from the chromatic vehicle class). Then, the probability of a pixel p belonging to the gray class is defined as follows:

$$P(Gray|p) = \exp\left(-\frac{(d_{mean} - u_{mean})^2}{\sigma_{mean}^2}\right) \exp\left(-\frac{(d_p - u_d)^2}{\sigma_d^2}\right), \quad (13)$$

where σ_{mean}^2 and σ_d^2 are the variances of d_{mean} and d_p , respectively. The following rule is used for determining whether p is a gray pixel:

$$P(Gray|p) > T_{gray}, \quad (14)$$

where T_{gray} is the threshold determined by averaging all gray pixels of vehicles during the training stage. In Fig. 17, (a) shows a vehicle image and (b) is the detection result for the gray pixels (denoted by in red) performed using Eq. (14). Given a vehicle \bar{R} , if the fraction of gray pixels in \bar{R} is greater than 80%, \bar{R} is classified into the “gray” class; otherwise, it belongs to the “color” class. Thus, the vehicle shown in Fig. 17(a) is classified as “color.”

D. SVM-Based Vehicle Classifiers

In the previous section, a G-classifier for classifying vehicles into “gray” and “color” classes was proposed. This section uses the SVM learning algorithm [21] to train the DG-classifier and DC-classifier for classifying the above classes. In Fig. 18, the DC-classifier attempts to classify the “color” vehicles into four categories, “red”, “green”, “blue”, and “yellow”. To train this classifier, thirty four features are

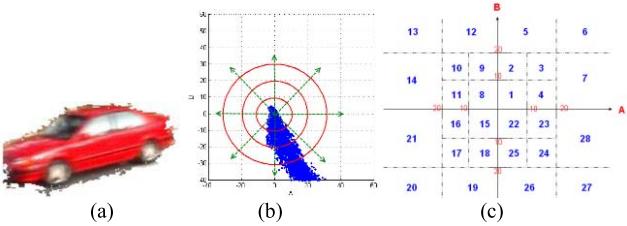


Fig. 19. Feature extraction from A - B color plane for vehicle classification. (a) Input frame. (b) Color mapping onto the A - B plane. (c) 28 dominant colors extracted for color classification.

extracted from both the LAB color space and the RGB color space. Here, twenty-eight features are extracted from the LAB domain and the remaining six features are extracted from the RGB color space. Because the LAB space separates the color components (A , B) of a pixel from its intensity component L , we choose LAB space for vehicle analysis. The separation can make the color components become more stable on an object even if small lighting changes or shadows exist. Because L channel is not related to color, only A and B channels are adopted for color classification. In Fig. 19, (a) is the original frame and (b) is the result of projecting (a) on the A - B plane. Clearly, different vehicle colors produce different distributions on the A - B plane. Then, we use polar coordinates to sample the A - B plane, where 10 units are used to quantize the radius and the angular direction is quantized into sections of width 90° . The sampling rate is non-linear. In Fig. 19 (c), each quadrant includes seven features. Twenty-eight grids are extracted from the polar coordinate. Each grid corresponds to a color bin. Then, we can construct a vector $h_{Lab} = (h_{Lab}(1), \dots, h_{Lab}(k), \dots, h_{Lab}(28))$, in which $h_{Lab}(k)$ is the number of pixels in the k th bin, *i.e.*,

$$h_{Lab}(k) = \#\{q | q \in bin^k\}, \quad (15)$$

where bin^k is the k th bin of the polar coordinates.

In real cases, a vehicle color is usually a combination of ‘on’ or ‘off’ selections in the RGB channels. For example, the ‘yellow’ color is a combination of ‘on’ flags in the RG channels and an ‘off’ flag in the B channel. The ‘on’ or ‘off’ values for a vehicle color can be useful features for vehicle color classification. Thus, we use the distributions of pixels whose color channels are larger than other channels in the RGB space as the other six features. For example, the 29th feature records the number of pixels for which the value of the B channel is larger than the G channel. Table 1 lists all six combinations of color comparisons. Each combination corresponds to a different feature. Then, a vector feature $h_{RGB} = (h_{RGB}(1), \dots, h_{RGB}(6))$ is extracted from the RGB color space. After integration, a feature vector h_{DC} with 34 dimensions is formed, $h_{DC} = (h_{Lab}, h_{RGB})$. Based on this set of features, the DC-classifier is trained using the SVM training algorithm.

To train the DG-classifier, we quantize each color channel in the RGB space into eight levels. Because three color channels are used, a vector h_{quan} with twenty-four features is extracted from this space. In addition, we record the

TABLE I
SIX COMBINATIONS OF COLOR COMPARISON. EACH COMBINATION CORRESPONDS TO A NEW FEATURE

Feature No.	29	30	31	32	33	34
Conditions	$B > G$	$G > R$	$G > B$	$G > R$	$R > B$	$R > G$

distributions of pixels whose color channels are larger than other channels. Then, as for the DC-classifier, a vector feature h_{RGB} is extracted. After integration, a feature vector h_{DG} with 30 dimensions is formed, $h_{DG} = (h_{quan}, h_{RGB})$. With h_{DG} , the DG-classifier can also be trained using the SVM algorithm.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed method, an automatic system for vehicle classification was implemented. Seven vehicle appearance categories (blue”, “green”, “red”, “yellow”, “white”, “silver”, and “black”) were used for performance evaluation. To train the DG-classifier and the DC-classifier, a database containing 3373 vehicles, in which 313 images are of white vehicles, 679 images are of black vehicles, 801 images are of silver vehicles, 404 images are of red vehicles, 465 images are of green vehicles, 263 images are of blue vehicles, and 448 images are of yellow vehicles, was used. Trucks are not included and tested in this paper because different advertisement text or logos with various colors that are painted on trucks can confuse the color classification. For training, the SVM library from [21] was used. The radial basis function (RBF) was selected as the kernel function; the gamma value used is 0.08125, and the parameter C of nu-SVC is 12.0. In addition, another test database containing images of 16,648 vehicles, in which 3,243 images are of white vehicles, 2783 images are of black vehicles, 2234 images are of silver vehicles, 1832 images are of red vehicles, 2303 images are of green vehicles, 2523 images are of blue vehicles, and 1730 images are of yellow vehicles, was also used. This vehicle database was collected by the VBIE (Vision-Based Intelligent Environment) project [28] from different roads, lighting (recording range from 6AM to 6PM), and weather conditions (sunny, cloudy, and raining). Now, the dataset is available from [30].

To theoretically analyze the performance of our proposed color correction scheme, this paper uses the “Fisher criterion” [22] to evaluate the separation ability of the vehicle classification method before and after color correction. The criterion uses the ratio of the “between-class” variance to the “within-class” variance to measure how well a transform T can separate a space into two classes C_1 and C_2 . The “between-class” variance is the distance between the classes’ means (denoted by m_1 and m_2 , respectively). The “within-class” variance is the sum of the classes’ variances, s_1 and s_2 . Then, the Fisher criterion is defined by

$$J(T) = \frac{\text{between-class distance}}{\text{within-class distance}} = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}.$$

For a color transform T , a larger value of J corresponds to a better separation ability. Table 2 lists the separation-

TABLE II
SEPARATION ABILITY ANALYSIS AMONG DIFFERENT COLOR
APPEARANCE CATEGORIES OF VEHICLE BEFORE
COLOR CORRECTION

Types	Red	Green	Blue	Yellow	Silver	Black	White
Red	X	0.041	0.023	0.031	0.038	0.010	0.017
Green	0.041	X	0.032	0.019	0.041	0.061	0.050
Blue	0.023	0.032	X	0.014	0.031	0.020	0.023
Yellow	0.031	0.019	0.014	X	0.030	0.038	0.028
Silver	0.038	0.041	0.031	0.030	X	0.032	0.034
Black	0.010	0.061	0.020	0.038	0.032	X	0.035
White	0.017	0.050	0.023	0.028	0.034	0.035	X

TABLE III
SEPARATION ABILITY ANALYSIS AMONG DIFFERENT COLOR
APPEARANCE CATEGORIES OF VEHICLE
AFTER COLOR CORRECTION

Types	Red	Green	Blue	Yellow	Silver	Black	White
Red	X	0.072	0.071	0.154	0.091	0.031	0.032
Green	0.072	X	0.043	0.108	0.121	0.075	0.065
Blue	0.071	0.043	X	0.072	0.059	0.080	0.041
Yellow	0.154	0.108	0.072	X	0.73	0.134	0.137
Silver	0.091	0.121	0.059	0.73	X	0.080	0.092
Black	0.031	0.075	0.08	0.134	0.080	X	0.061
White	0.032	0.065	0.041	0.137	0.092	0.061	X

ability analysis for the different vehicle color categories when the classification was performed before color correction. Table 3 shows the same analysis except that the classification was performed after color correction. Clearly, after color correction, the value of J is larger; this result implies that vehicles were classified more accurately after the color correction was performed. For comparison, the major color spectrum histogram representation (MCSHR) [7] was also implemented. This method is similar to a histogram-stretching technique for minimizing the color differences between two cameras. Fig. 20 shows a comparison of the results of color correction using our method and using the MCSHR method. (b) and (c) are the correction results for (a) using our method and the MCSHR method, respectively. Clearly, the colors of this vehicle were over-stretched by the MCSHR method. Fig. 21 shows another comparison of the results of our method and the MCSHR method. Table 4 gives the results of a separation-ability analysis after color correction using the MCSHR scheme. Comparison with Table 3 clearly demonstrates that our method performs better than the MCSHR scheme, especially for the categories “white” and “silver”.

In addition to the MCSHR method, the Reinhard’s method was also compared in this paper. Here, Eq.(1) and the aggregation of the estimated regression functions (see Eq.(9)) were integrated together for this comparison. From the comparison with Table 5, our method is more superior than the Reinhard’s method. To examine the effect of image segmentation, the mean-shift segmentation [20] method was also adopted in this paper. Table 6 shows the separation ability analysis using the mean-shift algorithm. Actually, the segmentation problem is still ill-posed now. Thus, compared

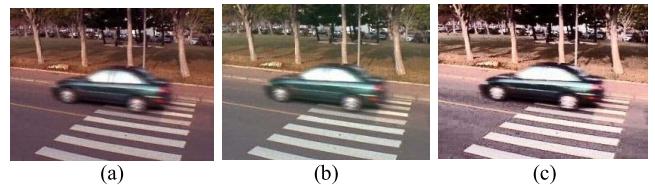


Fig. 20. Comparison results of color correction using our method and the MCSHR scheme [7]. (a) Input frame. (b) Result of color correction using our method. (c) Result of color correction using the MCSHR scheme.

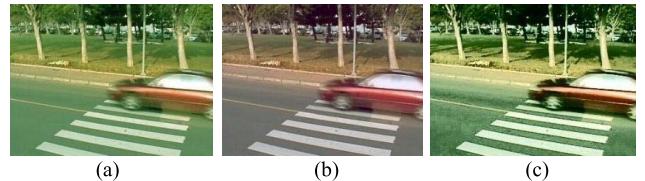


Fig. 21. Comparison results of color correction using our method and the MCSHR scheme [7]. (a) Input frame. (b) Result of color correction using our method. (c) Result of color correction using the MCSHR scheme.

TABLE IV
SEPARATION ABILITY ANALYSIS AMONG DIFFERENT COLOR
APPEARANCE CATEGORIES AFTER COLOR CORRECTION
USING THE MCSHR SCHEME [7]

Types	Red	Green	Blue	Yellow	Silver	Black	White
Red	X	0.027	0.026	0.016	0.041	0.014	0.021
Green	0.027	X	0.016	0.005	0.037	0.062	0.033
Blue	0.026	0.016	X	0.004	0.029	0.044	0.042
Yellow	0.016	0.005	0.004	X	0.021	0.055	0.043
Silver	0.041	0.037	0.029	0.021	X	0.041	0.082
Black	0.014	0.062	0.044	0.055	0.041	X	0.054
White	0.021	0.033	0.042	0.043	0.082	0.054	X

TABLE V
SEPARATION ABILITY ANALYSIS AMONG DIFFERENT COLOR
APPEARANCE CATEGORIES AFTER COLOR CORRECTION
USING REINHARD’S METHOD (SEE EQ.(1))

Types	Red	Green	Blue	Yellow	Silver	Black	White
Red	X	0.067	0.066	0.151	0.088	0.025	0.027
Green	0.067	X	0.04	0.098	0.091	0.069	0.062
Blue	0.066	0.04	X	0.066	0.051	0.075	0.04
Yellow	0.152	0.098	0.66	X	0.68	0.125	0.129
Silver	0.088	0.091	0.051	0.68	X	0.076	0.086
Black	0.025	0.069	0.075	0.125	0.076	X	0.055
White	0.027	0.062	0.041	0.129	0.086	0.055	X



Fig. 22. Results of vehicle window removal. (a) and (c) Input vehicles. (b) and (d) Results of window removal.

with Table 3, the mean-shift method performs better in some cases but worse in other cases.

In addition to the separation-ability analysis, we also examine the effects of vehicle window removal. Fig. 22 shows the results of vehicle window removal. (a) and (c) are the input

TABLE VI

SEPARATION ABILITY ANALYSIS AMONG DIFFERENT COLOR
APPEARANCE CATEGORIES AFTER COLOR CORRECTION
USING MEAN-SHIFT SEGMENTATION METHOD [20]

Types	Red	Green	Blue	Yellow	Silver	Black	White
Red	X	0.070	0.069	0.158	0.093	0.031	0.034
Green	0.070	X	0.054	0.105	0.108	0.073	0.071
Blue	0.069	0.054	X	0.066	0.053	0.077	0.042
Yellow	0.158	0.105	0.66	X	0.78	0.129	0.132
Silver	0.093	0.108	0.053	0.78	X	0.079	0.091
Black	0.031	0.073	0.077	0.129	0.079	X	0.058
White	0.034	0.071	0.042	0.132	0.091	0.058	X

TABLE VII

ACCURACY ANALYSIS OF VEHICLE CLASSIFICATION
BEFORE VEHICLE WINDOW REMOVAL

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	79.53	11.78	0.00	0.00	0.0	3.98	4.71
Silver	3.75	81.27	9.74	0.00	0.00	2.75	2.50
White	0.00	18.85	77.64	0.00	0.00	2.24	1.28
Yellow	5.80	9.60	0.00	83.71	0.67	0.22	0.00
Red	3.47	9.41	0.00	0.50	85.4	0.25	0.99
Green	8.39	10.97	0.00	0.00	0.00	76.34	4.30
Blue	10.65	13.31	0.00	0.00	0.00	4.56	71.48

TABLE VIII

ACCURACY ANALYSIS OF VEHICLE CLASSIFICATION USING THE
GRAY-CLASSIFIER BEFORE WINDOW REMOVAL

Categories	Gray vehicle	Color vehicle
Gray vehicle	8260	1945
Color Vehicle	0	6443
Accuracy	100%	76.81%

TABLE IX

ACCURACY ANALYSIS OF VEHICLE CLASSIFICATION USING THE
GRAY-CLASSIFIER AFTER WINDOW REMOVAL

Categories	Gray vehicle	Color vehicle
Gray vehicle	8193	117
Color Vehicle	67	8271
Accuracy	99.19%	98.61%

vehicles. (b) and (d) are the results of window removal for (a) and (c), respectively. Table 7 shows an accuracy analysis of the vehicle classification before vehicle window removal was performed. After vehicle window removal, the fraction of vehicle pixels with the same color increases, which yields significant improvements in the vehicle classification accuracy. The following set of experiments was used to evaluate the performance of the G-classifier to classify vehicles as “gray” or “color” before and after window removal. Table 8 shows the accuracy analysis of the G-classifier for classifying vehicles as “gray” or “color” before window removal. The accuracy for the “gray” category is 100% because the colors of vehicle windows naturally belong to the “gray” class and thus do not lead to errors. However, the third column of the table indicates that there are many color vehicles misclassified as “gray” if their vehicle windows were not removed. Table 9 shows the

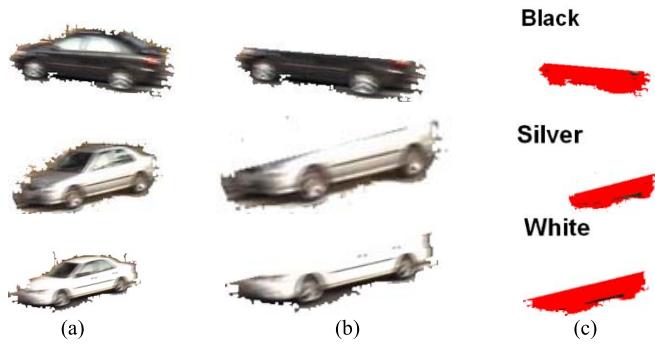


Fig. 23. Results of vehicle classification using the DG-classifier. (a) Input vehicles. (b) Results of vehicle window removal. (c) Results of vehicle classification where the red color shows the detected gray pixels using Eq. (14).

TABLE X
CONFUSION MATRIX OF VEHICLE CLASSIFICATION USING THE
DG-CLASSIFIER BEFORE VEHICLE WINDOW REMOVAL

Types	Black	Silver	White	Black-G	Black-B
Black	86.89%	7.82%	0.00%	2.28%	3.01%
Silver	7.49%	78.23%	9.74%	2.31%	2.23%
White	0.00%	14.68%	85.32%	0.00%	0.00%
Black-G	8.39%	7.66%	0.00%	81.73%	2.22%
Black-B	7.95%	4.98%	0.00%	4.73%	82.34%

TABLE XI
CONFUSION MATRIX OF VEHICLE CLASSIFICATION USING THE
DG-CLASSIFIER AFTER VEHICLE WINDOW REMOVAL

Types	Black	Silver	White	Black-G	Black-B
Black	92.74%	5.28%	0.00%	1.15%	0.83%
Silver	4.88%	85.99%	7.65%	0.72%	0.76%
White	3.36%	3.70%	92.94%	0.00%	0.00%
Black-G	5.21%	2.17%	0.00%	84.89%	2.41%
Black-B	6.34%	0.99%	0.00%	4.52%	87.01%

same analysis of the G-classifier but after window removal was performed. The third column indicates that the accuracy of color classification was significantly improved by the window removal (from 76.81% to 98.61%). However, the accuracy is still lower for the color class than for the gray class because some blue or green vehicles are similar to gray vehicles. The average accuracy of the G-classifier is 99.19%. Clearly, the accuracy of the G-classifier is sufficient for achieving a highly accurate vehicle classification.

After analyzing the performance of the Gray-classifier, we evaluate the performance of the DG-classifier and the DC-classifier. Fig. 23 shows classification results of the DG-classifier. (b) shows the results of vehicle window removal from (a). (c) shows the results of vehicle classification of (b). The red area denotes the gray pixels detected using Eq. (14). Table 10 shows the confusion matrix of the DG-classifier before window removal.

Table 11 shows the same analysis but after window removal. Some vehicles in the “silver” category are very similar to those in the “white” category. Thus, they were misclassified into the “white” category. In addition, the “navy” and “deep green” vehicles were often misclassified into the “black” category.

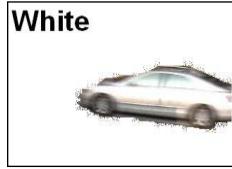


Fig. 24. Failure case of the DG-classifier of a “silver” vehicle to be “white”.

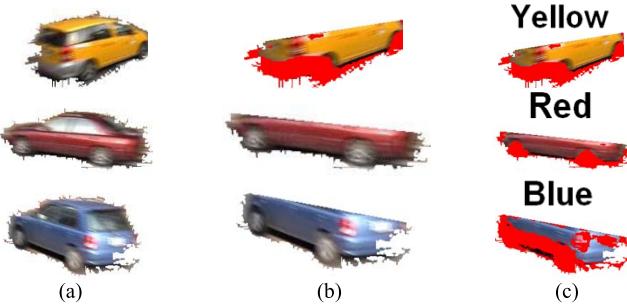


Fig. 25. Results of vehicle classification using the DC-classifier. (a) Input vehicles. (b) Results of vehicle window removal. (c) Results of classification where the red color shows the detected gray pixels using Eq.(14).

TABLE XII

CONFUSION MATRIX OF VEHICLE CLASSIFICATION USING THE DC-CLASSIFIER BEFORE VEHICLE WINDOW REMOVAL

Types	Yellow	Red	Green	Blue
Yellow	97.11%	2.29%	0.60%	0.00%
Red	1.07%	97.92%	0.48%	0.53%
Green	0.00%	0.00%	93.54%	6.46%
Blue	0.00%	0.00%	9.40%	90.60%

TABLE XIII

CONFUSION MATRIX OF VEHICLE CLASSIFICATION USING THE DC-CLASSIFIER AFTER VEHICLE WINDOW REMOVAL

Types(%)	Yellow	Red	Green	Blue
Yellow	99.10	0.32	0.45	0.13
Red	0.00	99.34	0.25	0.41
Green	0.00	0.00	98.52	1.48
Blue	0.00	0.00	4.78	95.22

Table 9 and Table 11 clearly indicate that window removal can significantly improve the accuracy of classification. Fig. 24 shows a case in which our method failed to classify a silver vehicle. The effect of specular highlights made the colors of the vehicle windows and body appear white, which caused our method to fail.

Fig. 25 shows the results of vehicle classification using the DC-classifier. (b) shows the results of vehicle window removal from (a). In (c), the gray pixels (denoted in red) were first detected using Eq. (14). Then, all the non-gray pixels were retained for vehicle classification using the DC-classifier. The top captions in (c) show the results of the vehicle classification. Table 12 shows the confusion matrix of the DC-classifier before window removal. Table 13 shows the same analysis but after window removal. In real cases, the “red” and “yellow” categories are more easily classified because they are more vivid than the blue



Fig. 26. Results of vehicle classification when the road included multiple vehicles. (a) Two vehicles. (b) Four vehicles.



Fig. 27. Some false recognition results. (a) False recognition of a blue vehicle. (b) False recognition of a green vehicle.



Fig. 28. Some false recognition results from front views. (a) False recognition of a silver vehicle to be “white”. (b) False recognition of a blue vehicle to be “dark green”.

TABLE XIV
CONFUSION MATRIX OF VEHICLE CLASSIFICATION BEFORE COLOR CORRECTION (%)

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	86.24	5.28	0.00	0.00	0.00	4.02	4.10
Silver	6.62	77.04	10.83	0.00	0.00	2.51	3.00
White	3.27	9.56	86.77	0.00	0.00	0.00	0.40
Yellow	0.58	0.75	0.29	98.38	0.00	0.00	0.00
Red	1.58	0.27	0.00	0.00	98.14	0.00	0.00
Green	8.68	3.47	0.00	0.00	0.00	82.50	5.34
Blue	11.89	2.58	0.00	0.00	0.0%	3.49	82.05

and green colors. When the lighting conditions are poor, the blue and green colors appear very similar to “black”. Thus, higher accuracies were obtained for the “red” and “yellow” categories. Fig. 26 shows the results of vehicle classification performed on an image of a real road that includes multiple vehicles. Fig. 27 shows two cases in which the DC-classifier failed. Fig. 28 shows another set of failure cases caused by shade colors. The effects of specular highlights caused these vehicles to be misclassified as a fake label.

In addition to analyzing the effects of window removal, we also evaluate the accuracy improvement of vehicle classification yielded by performing color correction. Table 14 gives the confusion matrix of vehicle classification before color correction. The average accuracy of the vehicle classification is approximately 86.58%. Table 15 shows the same accuracy analysis but after color correction without region segmenta-

TABLE XV

CONFUSION MATRIX OF VEHICLE CLASSIFICATION AFTER COLOR
CORRECTION WITHOUT REGION SEGMENTATION

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	92.74	5.17	0.00	0.00	0.00	1.11	0.97
Silver	4.83	86.03	7.48	0.00	0.00	0.81	0.85
White	3.36	3.70	92.94	0.00	0.00	0.00	0.00
Yellow	0.81	0.29	0.00	98.90	0.00	0.00	0.00
Red	1.86	0.44	0.00	0.00	97.71	0.00	0.00
Green	5.21	2.34	0.00	0.00	0.00	92.40	0.04
Blue	6.38	0.95	0.00	0.00	0.00	2.73	89.93

TABLE XVI

CONFUSION MATRIX OF VEHICLE CLASSIFICATION AFTER COLOR
CORRECTION WITH REGION SEGMENTATION

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	94.50	4.20	0.00	0.00	0.00	0.57	0.72
Silver	3.54	89.12	6.31	0.00	0.00	0.49	0.54
White	2.7	3.70	93.56	0.00	0.00	0.00	0.00
Yellow	0.29	0.40	0.00	99.31	0.00	0.00	0.00
Red	0.87	0.27	0.00	0.00	98.85	0.00	0.00
Green	4.73	2.21	0.00	0.00	0.00	93.01	0.04
Blue	4.91	0.99	0.00	0.00	0.00	1.90	92.19

TABLE XVII

CONFUSION MATRIX OF VEHICLE CLASSIFICATION WHEN VEHICLES
WERE CAPTURED FROM HIGHWAYS

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	93.09	3.89	0.43	0.00	0.00	0.43	2.16
Silver	4.64	89.40	5.52	0.00	0.00	0.00	0.44
White	0.72	7.61	91.67	0.00	0.00	0.00	0.00
Yellow	0.93	0.69	0.00	98.15	0.00	0.00	0.23
Red	0.60	0.90	0.00	0.00	98.49	0.00	0.00
Green	5.10	1.21	0.00	0.00	0.00	89.32	4.37
Blue	9.48	3.02	0.00	0.00	0.00	1.29	86.21

tion was performed. The average accuracy is approximately 92.59%. Table 16 shows the analysis after color correction with region segmentation was performed (see Eq.(9)). Clearly, color correction can enhance the color differences between the “silver” category and other categories.

To further examine the robustness of our system, we also tested our method for various roads and weather conditions. Table 17 gives the confusion matrix of vehicle classification on highways. The average accuracy is 92.62%. The most difficult classification case is “silver” because it is very similar to “white”. Table 18 shows an analysis of vehicle classification for vehicles in different parking spaces. The average accuracy is 91.56%. The numbers of vehicles tested in the highway dataset and the parking space dataset are 9,632, and 7,016, respectively. The two tables indicate that better results were achieved for the “red” and “yellow” categories because of their vivid colors. Weather is another important factor that affects the accuracy of vehicle classification. Table 19 shows the confusion matrix of vehicle classification for cloudy days. The average accuracy is 94.07%. Clearly, on a cloudy day, better performance is obtained because the sunlight is weaker.

TABLE XVIII

CONFUSION MATRIX OF VEHICLE CLASSIFICATION WHEN VEHICLES
WERE CAPTURED FROM PARKING SPACES

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	91.52	6.14	0.58	0.00	0.00	0.58	2.92
Silver	3.97	88.74	5.74	0.00	0.00	0.00	1.55
White	0.86	10.34	91.38	0.00	0.00	0.00	0.00
Yellow	2.18	1.25	0.00	98.75	0.00	0.00	0.00
Red	0.70	1.05	0.00	0.00	98.26	0.00	0.00
Green	5.56	1.54	0.00	0.00	0.00	89.20	3.70
Blue	6.63	2.11	0.00	0.00	0.00	0.90	90.36

TABLE XIX

CONFUSION MATRIX OF VEHICLE CLASSIFICATION WHEN VEHICLES
WERE CAPTURED UNDER CLOUDY DAYS

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	93.43	1.73	1.73	0.00%	0.00	1.73	1.38
Silver	4.66	89.86	4.93	0.00	0.00	0.55	0.00
White	1.46	5.85	92.11	0.00%	0.00	0.29	0.29
Yellow	0.65	0.32	0.00	99.03	0.00	0.00	0.00
Red	0.90	0.30	0.00	0.00	98.80	0.00	0.00
Green	3.70	3.70	0.00	0.00	0.00	92.59	0.00
Blue	3.61	3.01	0.00	0.00	0.00	0.00	93.37

TABLE XX

CONFUSION MATRIX OF VEHICLE CLASSIFICATION WHEN A
SECTOR-SAMPLING TECHNIQUE WAS ADOPTED

Types	Black	Silver	White	Yellow	Red	Green	Blue
Black	92.24	5.68	0.00	0.00	0.00	1.22	0.86
Silver	4.03	90.11	4.48	0.00	0.00	0.63	0.76
White	3.39	4.07	92.54	0.00	0.00	0.00	0.00
Yellow	0.87	0.64	0.00	98.50	0.00	0.00	0.00
Red	1.58	0.22	0.00	0.00	98.20	0.00	0.00
Green	5.25	2.30	0.00	0.00	0.00	92.40	0.04
Blue	6.42	0.91	0.00	0.00	0.00	1.51	91.16

In Section III.C, the polar coordinate (see Fig. 19(c)) is adopted for sampling some important color bins for vehicle classification. Table 20 shows an accuracy analysis of vehicle classification using a sector-type sampling on the polar coordinates (see Fig. 19(b)). The average accuracy is 93.18%. Table 16 shows another confusion matrix of vehicle classification when the rectangular sampling shown in Fig. 19(c) was adopted. The average accuracy is 93.59%. Clearly, the proposed rectangle-type sampling technique is superior to the uniform sample scheme. Another advantage of this sampling technique is an improvement in efficiency because sampling on the A-B color space using rectangular blocks is easier than sampling using sectors.

For comparison, four methods [24]–[27] for vehicle classification were implemented. In [24], the back propagation network was adopted to classify vehicles via container separation from the HSI color space. Its performances strongly depend on a proper selection of thresholds used in “container” classification. In [25], the success of color classification majorly depends on the detections of vehicle taillights. Their method is instable to separate the category “dark blue” from “dark

TABLE XXI
ACCURACY COMPARISONS AMONG FOUR METHODS [24]–[27]

Types(%)	Black	S	W	Y	Red	G	Blue	Avg.
Fang [24]	69.52	72.53	76.92	79.45	84.12	62.59	61.34	72.35
Wang [25]	73.82	75.45	78.43	83.23	84.82	65.57	62.72	74.86
Yang [26]	65.82	73.56	68.69	76.78	80.21	55.25	47.83	66.88
Baek [27]	57.02	66.34	57.42	75.26	76.42	51.15	43.64	61.04
Our Meth.	92.24	90.11	92.54	98.50	98.20	92.40	91.16	93.59

green". As to [26], a rule-based classifier was proposed to classify colorful vehicles into different category based on the RGB and HSV color space. The used rule-based classifier is easily degraded by the effects of sunlight because the lighting conditions are out of the defined rules. In [27], a brute-force scheme was proposed to divide the hue and saturation values of the HSV color space into 360 elements for training a vehicle classifier. This method is not stable when the lighting has changes. Because the vehicle window is remained, its accuracy is also degraded by the effects of sunlight. Table 21 presents the accuracy comparisons among our proposed method and other methods [24]–[27]. The average accuracy of our method is 93.59%. Clearly, our vehicle classification method performs much better than other methods. Experimental results have demonstrated the superiority of our method for vehicle classification.

V. CONCLUSION

We have proposed a novel method for classifying vehicles to different color categories. The contributions of this method are summarized as follows:

- 1) A novel color correction scheme was proposed for reducing the effect of lighting change.
- 2) A novel window-removal scheme was proposed for removing the effect of sun lighting.
- 3) A novel tree-based classifier was designed for classifying vehicles into different categories even from different cameras and under different lighting conditions.
- 4) A rectangular block sampling technique was proposed for sampling important features from the A-B color space for vehicle representation.

The average accuracy of vehicle classification with our method was 93.59%. The experimental results demonstrate that the proposed method is superior to other vehicle classification methods in terms of accuracy, robustness, and stability.

REFERENCES

- [1] R. T. Collins *et al.*, "A system for video surveillance and monitoring," Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-00-12, May 2000.
- [2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006, Art. ID 13.
- [3] K.-W. Chen, C.-C. Lai, P.-J. Lee, C.-S. Chen, and Y.-P. Hung, "Adaptive learning for target tracking and true linking discovering across multiple non-overlapping cameras," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 625–638, Aug. 2011.
- [4] C. Picus, R. Pfugfelder, and B. Micusik, "Branch and bound global optima search for tracking a single object in a network of non-overlapping cameras," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 1825–1830.
- [5] G. Lian, J. Lai, and W. Zheng, "Spatial-temporal consistent labeling of tracked pedestrians across non-overlapping camera views," *Pattern Recognit.*, vol. 44, no. 5, pp. 1121–1136, May 2011.
- [6] G. Lian, J.-H. Lai, C. Y. Suen, and P. Chen, "Matching of tracked pedestrians across disjoint camera views using CI-DLBP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1087–1099, Jul. 2012.
- [7] C. Madden, E. D. Cheng, and M. Piccardi, "Tracking people across disjoint camera views by an illumination-tolerant appearance representation," *Mach. Vis. Appl.*, vol. 18, nos. 3–4, pp. 233–247, May 2007.
- [8] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun./Jul. 2004, pp. II-205–II-210.
- [9] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Spectral sharpening: Sensor transformations for improved color constancy," *J. Opt. Soc. Amer. A*, vol. 11, no. 5, pp. 1553–1563, May 1994.
- [10] G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.*, vol. 310, no. 1, pp. 1–26, 1980.
- [11] J. van de Weijer and T. Gevers, "Color constancy based on the grey-edge hypothesis," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2005, pp. II-722–II-725.
- [12] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [13] Y. Xiang, B. Zou, and H. Li, "Selective color transfer with multi-source images," *Pattern Recognit. Lett.*, vol. 30, no. 7, pp. 682–689, May 2009.
- [14] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakaushi, "Illumination normalization with time-dependent intrinsic images for video surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1336–1347, Oct. 2004.
- [15] K. Kim, T. H. Chalidabongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, 2005.
- [16] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.
- [17] Y.-W. Tai, J. Jia, and C.-K. Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 747–754.
- [18] M. Oliveira, A. D. Sappa, and V. Santos, "Unsupervised local color correction for coarsely registered images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 201–208.
- [19] P. F. Felzenszwalb and D. P. Huttenlocher, "Image segmentation using local variation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 98–104.
- [20] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [21] C.-C. Chang and C.-J. Lin. (2001). LIBSVM: A Library for Support Vector Machines. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [22] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.
- [23] L. M. Brown, "Example-based color vehicle retrieval for surveillance," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug./Sep. 2010, pp. 91–96.
- [24] J. Fang, H. Yue, X. Li, J. Wu, and Z. Cui, "Color identifying of vehicles based on color container and BP network," in *Proc. Int. Conf. Bus. Manage. Electron. Inf.*, May 2011, pp. 226–229.
- [25] Y.-C. Wang, C.-T. Hsieh, C.-C. Han, and K.-C. Fan, "The color identification of automobiles for video surveillance," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol. (ICCSST)*, Oct. 2011, pp. 1–5.
- [26] M. Yang, G. Han, X. Li, X. Zhu, and L. Li, "Vehicle color recognition using monocular camera," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Nov. 2011, pp. 1–5.
- [27] N. Baek, S.-M. Park, K.-J. Kim, and S.-B. Park, "Vehicle color classification based on the support vector machine method," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques* (Communications in Computer and Information Science), vol. 2, Berlin, Germany: Springer-Verlag, 2007, pp. 1133–1139.
- [28] VIE Web, [Online]. Available: <http://vbie.eic.nctu.edu.tw/en/introduction>, accessed Aug. 2008.
- [29] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 6–20, Feb. 2014.
- [30] NTOU Vehicle Dataset, [Online]. Available: <http://mmlab.cs.ntou.edu.tw/MMR/>, accessed Sep. 2013.



Jun-Wei Hsieh (M'06) received the bachelor's degree in computer science from Tunghai University, Taichung, Taiwan, in 1990, and the Ph.D. degree in computer engineering from National Central University, Zhongli, Taiwan, in 1995. He was a recipient of the Phai-Tao-Phai Award. From 1996 to 2000, he was a Research Fellow with the Industrial Technology Research Institute, Hsinchu, Taiwan, where he managed a team to develop video-related technologies. He was an Assistant Professor with the Department of Electrical Engineering, Yuan Ze University, Zhongli, Taiwan, from 2001 to 2003, and an Associate Professor from 2004 to 2009. He was the Chairman and is currently a Professor with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan. He received the Best Paper Awards from the Image Processing and Pattern Recognition Society of Taiwan in 2005, 2006, and 2007, respectively. He was a recipient of the Best Patent Award from the Industrial Technology Research Institute of Taiwan in 2009, and the best paper awards from IIHMSP 2010, DMS2011, ITOAI 2013, ITOAI 2014, and CVGIP 2014, respectively. His research interests include content-based multimedia databases, video surveillance, intelligent transportation system, computer vision, and pattern recognition.



Li-Chih Chen received the B.S. degree in information science from Chinese Culture University, Taipei, Taiwan, in 1992, and the M.S. degree in information system from Lawrence Technological University, Southfield, MI, USA, in 1999. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering, Yuan Ze University, Zhongli, Taiwan. He is currently a Lecturer with the Department of Electrical Engineering, Lee-Ming Institute of Technology, Taipei.



Sin-Yu Chen was born in Taipei, Taiwan, in 1982. He received the B.S. and Ph.D. degrees in electrical engineering from Yuan Ze University, Zhongli, Taiwan, in 2005 and 2011, respectively. His research interests include image processing, behavior analysis, and video surveillance.



Duan-Yu Chen received the B.S. degree in computer science and information engineering from National Chaio Tung University, Hsinchu, Taiwan, in 1996, the M.S. degree in computer science from National Sun Yat-sen University, Kaohsiung, Taiwan, in 1998, and the Ph.D. degree in computer science and information engineering from National Chiao Tung University in 2004. He was a Post-Doctoral Research Fellow with Academia Sinica, Taipei, from 2004 to 2008. He is currently an Associate Professor with the Department of Electrical Engineering, Yuan Ze University, Zhongli, Taiwan. His research interests include computer vision, video signal processing, content-based video indexing and retrieval, and multimedia information system. He was a recipient of the Young Scholar Research Award from Yuan Ze University in 2012.



Salah Alghyalline received the B.S. degree in computer science from Mutah University, Karak, Jordan, in 2005, the M.S. degree in computer information systems from the Arab Academy for Banking and Financial Sciences University, Amman, Jordan, in 2007. He is currently pursuing the Ph.D. degree at the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan.



Hui-Fen Chiang received the B.S. degree in computer science and information engineering from Feng Chia University, Taichung, Taiwan, in 1992, and the M.S. degree in computer science and information engineering from National Central University, Taoyuan, Taiwan, in 1995. She is currently pursuing the Ph.D. degree at National Taiwan Ocean University, Keelung, Taiwan. She is currently a Lecturer with the Department of Digital Multimedia Design, Taipei Chengshih University of Science and Technology, Taipei, Taiwan.