VEHICLE COLOR CLASSIFICATION UNDER DIFFERENT LIGHTING CONDITIONS THROUGH COLOR CORRECTION

Jun-Wei Hsieh[1]*, Li-Chih Chen*[2]*, Sin-Yu Chen*[2]*, Shih-Chun Lin*[1]*, and Duan Yu Chen*[2]
Department of Computer Science and Engineering,
[1]National Taiwan Ocean University, No.2, Beining Rd., Keelung 202, Taiwan
E-mail: shieh@ntou.edu.tw
[2]Department of Electrical Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li 320, Taiwan

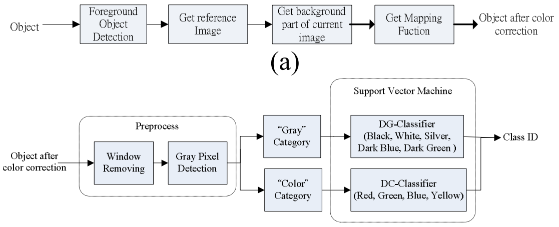
ABSTRACT

This paper presents a novel color correction technique for classifying vehicles under different lighting conditions using their colors. To reduce the lighting effects, a reference image is first selected for building the mapping function between the current frame and the reference image. *With this mapping function, the color distortions between frames can be reduced to minimum. In addition to lighting changes, the effect of sun light will make the vehicle window become white and lead to the errors of vehicle classification. To reduce this effect, a window-removing task is then applied for making vehicle pixels with the same color more concentrated on the foreground region. Then, vehicles can be more accurately classified to their categories even though strong sun light casts on them. To tackle the confusion problem that some vehicle colors are too similar, e.g., "deep-blue" and "deep-green", a novel tree-based classifier is then designed for classifying vehicles to more detailed labels. Experimental results have proved that the proposed method is a robust, accurate, and powerful tool for vehicle classification.*

1. INTRODUCTION

Video surveillance in public spaces has attracted immense attention in recent years due to its extreme capabilities in crime prevention and security maintenance. When a wide-area scene is monitored, due to the limited field of view (FOV) of a camera, existing surveillance systems usually adopt multiple cameras for smoothly object tracking and identification [1]-[7]. However, due to different lighting changes and camera distortions, the visual properties of an object will change significantly. Under these conditions, how to maintain the identity of a target moving across cameras has become very challenging. During the past few decades, extensive studies [2]-[7] have been conducted on object tracking and then identifying across multiple cameras. According to the FOV relations between cameras, the above frameworks can be divided into two categories, *i.e.*, overlapping and non-overlapping. For the first case, most approaches focused on building the feature correspondences within the overlapping area of cameras so that the inter-camera homography can be computed and then different objects can be well tracked. For example, Chang and Gong [4] used Bayesian Belief Network (BBN) models to probabilistically infer the correspondence relations of people from two cameras and then track them in an indoor environment. In [5], Khan and Shah solved the occlusion problem by using a homography constraint for tracking people in a multi-camera system. Compared with the case of overlapping cameras, establishing correspondences across multiple non-overlapping cameras is more challenging due to the lack of overlapping information between cameras. To tackle the problem, different space-time cues are used to build feature correspondences and then the camera relations can be associated. For example, in [2], Javed *et al.* proposed a learning algorithm to learn the subspace of inter-camera brightness transfer functions for correcting bright changes between cameras and then tracking objects. However, in real cases, the observations of the same object coming from unknown cameras change very different in time and space.

Maintaining the color constancy between cameras is another important technique for tackling the association problem among multiple cameras. For example, Finlayson, Fredembach, and Drew [6] used a chromagenic camera to capture two images simultaneously from the same scene and then proposed a matrix-based method to detect its multiple illuminants. In addition, Xiang, *et al.* [11] proposed a model-based transform between two images for reducing the color distortions between two cameras. However, the above methods will fail when the lighting conditions of the observed environment are not fixed.


Object
Foreground Object Detection
Get reference Image
Get background part of current image
Get Mapping Fuction
Object after color correction
Support Vector Machine
Preprocess
Object after color correction
Window Removing
Gray Pixel Detection
"Gray" Category
"Color" Category
DG-Classifier (Black, White, Silver, Dark Blue, Dark Green)
DC-Classifier (Red, Green, Blue, Yellow)
Class ID

(a)

(b)

Fig. 1 : Overview of the proposed system.

This paper presents a novel color correction technique to classify vehicles using their color features. Fig. 1 shows the flowchart of our proposed system. Usually, a vehicle's color will change under different conditions. To reduce the lighting variations and color changes, a color mapping function between a background image and current image is estimated. After this color mapping, the color distortions among different cameras

978-1-4673-0219-7/12/$31.00 ©2012 IEEE

can be reduced to minimum. In addition to lighting changes, the effect of sun light will make the vehicle window become white and degrade the accuracy of vehicle classification. To classify a vehicle more accurately, a window removing scheme is then proposed for making vehicle pixels with the same color more concentrated on the foreground region. After that, a novel tree-based classifier is designed for classifying vehicles from coarse to fine labels. At the root node, a G-classifier is designed for classifying vehicles to “gray” or “color” classes. Then, the root node is split to two child nodes named as the “color” node and the “gray” node for detailing the gray class and the color class. At each child node, a polar coordinate is used for non-linearly sampling the AB space for representing an object with a set of important color features. For the color-child, a feature vector with thirty four dimensions is constructed for training a DC-classifier (DC, “detailed color”) by using the SVM learning algorithm. For the gray-child node, a DG-classifier (DG, “detailed gray”) with thirty features is trained for classifying gray vehicles to more detailed categories. Experimental results demonstrate the feasibility and superiority of the proposed approach in object identification across different cameras.

2. COLOR COMPENSATION

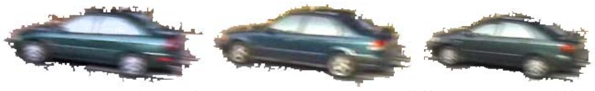


Fig. 2 : The colors of the same vehicle are different under different lighting conditions.

When observing a vehicle from different viewpoints, its colors will change at different time. Like Fig. 2, the colors of the same vehicle will change differently under different lighting conditions. Thus, it is quite important to correct the color changes before classification.

Actually, in [9], Reinhard *et al.* reported a simple technique that transfers color characteristics from a source I_s to a target image I_t. Let μ_s and μ_t be the means of I_s and I_t with the variances σ_s and σ_t, respectively. Then, a mapping function $g(p)$ for transferring the color of a pixel p from I_t to I_s is defined as:

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

In real implementations, Eq.(1) needs to scan I_s and I_t twice individually for calculating σ_s and σ_t and becomes inefficient in real time applications. We can rewrite Eq.(1) as the form

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

where $\alpha = \frac{\sigma_s}{\sigma_t}$ and $\beta = \mu_s - \mu_t$. 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:

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

Let N be the size of I_s. After calculations, we get

$$\alpha = \frac{N\mu_s\mu_t - \sum I_t I_s}{N\mu_t^2 - \sum I_t^2} \text{ and } \beta = \frac{\mu_t(\sum I_t I_s) - \mu_s \sum I_t^2}{N\mu_t^2 - \sum I_t^2}. \quad (4)$$

Fig. 3 shows one example of the result of color correction. (a) is the reference image and (b) is the input frame. (c) is the result of color correction from (b).

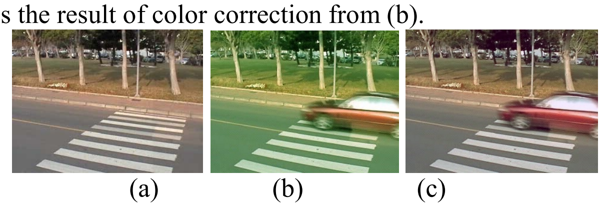


Fig. 3 : Results of color correction by the proposed mapping function.

3. VEHICLE IDENTIFICATION

In addition to the color change, the effect of sun light will also make the vehicle window become white and degrade the accuracy of classification (see Fig. 4). To reduce this effect, the vehicle window should be first removed.

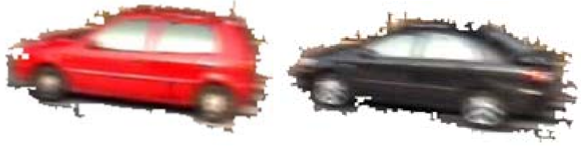


Fig. 4: Window colors became white due to the sun light.

3.1 Vehicle Window Removing

Let R denote the vehicle extracted by a background subtraction technique. To remove its window, the major orientation θ_R of R should be first detected. Let the central moment of R be defined as follows:

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

where $(\bar{x}, \bar{y})$ is the center of R with the area $|R|$. Then, the orientation θ_R of R can be estimated as

$$\theta_R = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right]. \qquad (5)$$

Let L_{cut} denote the line with the orientation θ_R and passing the center $(\bar{x}, \bar{y})$. For each pixel p in R, if p is below L_{cut}, p will be collected to the remained vehicle $\bar{R}$. Then, a tree-based classifier will be trained for classifying $\bar{R}$ to different categories.

3.2 Vehicle Categories

After window removing, the remained vehicle $\bar{R}$ will be further classified to different categories. There are seven categories collected in this paper for classification, *i.e.*, {“blue”, “green”, “red”, “yellow”, “white”, “silver”, “black”}. In real cases, for a vehicle with a “dark-blue” or “dark-green” color, although its category belongs to “blue” or “green”, it is often misclassified as “black”. For tackling this confusion problem, it is better to divide the “blue” and “green” classes to four categories, *i.e.*, {“blue”, “dark blue”, “green”, “dark green”}. Then, a tree structure is proposed for classifying vehicles from coarse to fine categories. Like Fig. 5, at the root node, a G-classifier (G for gray) is proposed for classifying vehicles only to two classes, *i.e.*, "gray" and "color". The "gray" class containing the {"black", "silver", "white", "dark-blue", "dark-green"} categories and the "color" class containing the {"red", "green", "blue", "yellow"} categories. Fig. 6 shows all the vehicle color categories. For the child nodes, the "DG-classifier" and "DC-classifier" are trained for classifying the "gray" and "color" classes to more detailed labels.

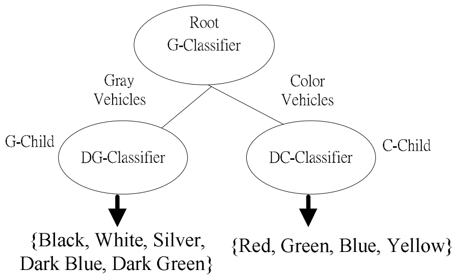
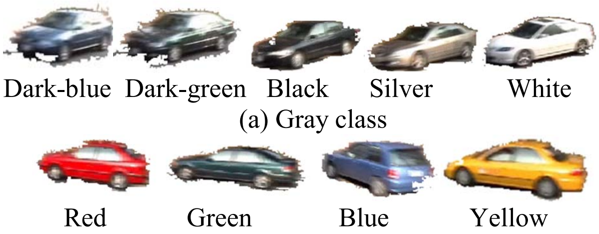


Fig. 5 : A tree structure for vehicle classification.



(a) Gray class

(b) Color class

Fig. 6 :Vehicles classified to "gray" and "color" classes.

3.3 Gray Pixel Identification

At the root node, a G-classifier is proposed for classifying vehicles only to two classes, *i.e.*, "gray" and "color". To build the G-classifier, we should decide the probability of a pixel p belonging to gray. If the color $(r,g,b)^t$ of p is gray, it should be very close to the gray axis $(1/3,1/3,1/3)^t$. Thus, we define the distance between p and the gray axis as follow:

$$d_p = (r-0.333)^2 + (g-0.333\ )^2 + (b-0.333\ )^2. \qquad (6)$$

Then, the probability of p to be gray is defined as

$$P(Gray\,|\,p) = \exp(-\frac{(d_p - u_m)^2}{\sigma_m^2}), \qquad (7)$$

where u_m and σ_m^2 are the mean and variance of d_p. For the G-classifier, the rule for determining whether p is a gray pixel is defined with the form:

$$P(\ Gray\ \ |\ \ p\ ) > g_{threshold}\ . \qquad (8)$$

Then, if the ratio of gray pixels in $\bar{R}$ is more than 80%, $\bar{R}$ is classified to the "gray" class.

3.4 SVM-based Vehicle Classification

This section uses the SVM learning algorithm [8] to train the DC-classifier and DG-classifier for classifying the "color" and "gray" classes to more detailed labels. To train the DC-classifier, thirty four features are extracted from the LAB color space and the RGB color space, respectively. Here, twenty eight features are extracted from the LAB domain and the remained six features are extracted from the RGB color space. We use a polar coordinate to sample the A-B plane, where 10 units are used to quantize the radius and 90° is used to quantize the angle. Like Fig. 8, (a) is the original frame and (b) is the projection result of (a) on the A-B plane. Then, twenty eight grids are extracted from the polar coordinate. Each grid corresponds to a color bin. Then, a 28-dimensional vector histogram is constructed, *i.e.*, h_{Lab} = ($h_{Lab}(1)$, ...,$h_{Lab}(k)$, …, $h_{Lab}(28)$), in which $h_{Lab}(k)$ is the number of pixels in the *k*th bin, i.e.,

$$h_{Lab}(k) = \#\ \{q\,|\ q\in\ bin^k\}\ , \qquad (9)$$

where bin^k is the *k*th bin of the polar coordinate. As to other six features extracted from the RGB domain, we record the distributions of pixels whose color channels are larger than other channels in the RGB space. There are six combinations of color comparisons. Then, a 6-dimentianl vector $h_{RGB} = (h_{RGB}(1),\ ...,h_{RGB}(6)\}$ can be extracted from the RGB color space. After integration, a feature vector h_{DC} with 34 dimensions is constructed, *i.e.*, $h_{DC} = (h_{Lab}, h_{RGB})$. Then, the DC-classifier can be trained using the SVM training algorithm [8].

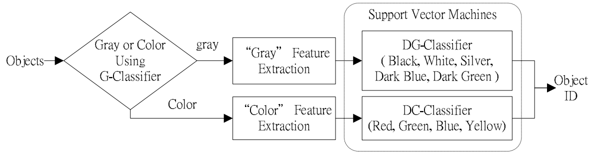


Fig. 7 : Two classifiers were trained for classifying vehicles to different categories.

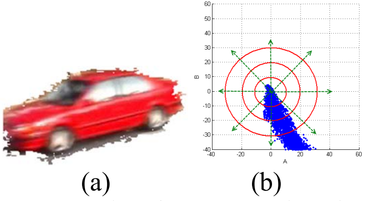


Fig. 8: Feature extraction from A-B color plane for vehicle classification. (a) Input frame. (b) Color mapping onto the A-B plane.

To train the DG-classifier, we quantize each color channel on the RGB space to eight levels. Since three color changes are used, a vector h_{quan} with twenty four features will be constructed. In addition, we record the distributions of pixels whose color channels are larger than other channels on the RGB space. After integration, a feature vector h_{DG} with 30 dimensions is formed, *i.e.*, $h_{DG} = (h_{quan}, h_{RGB})$. With h_{DG} , the DG-classifier is trained using the SVM training algorithm.

4. EXPERIMENTAL RESULTS

To theoretically analyze the performance of our proposed correction scheme, the "Fisher criterion" [10] is used to evaluate the separation ability of vehicle classification before and after color correction. The criterion $J(T)$ measures how well a transform T can separate a space into two classes. The larger value of J is, the better separation ability T has. Table 1 and Table 2 show the separation ability analyses among different vehicle categories before and after color correction. After comparisons, from the view of theoretical analysis, the improvement of color correction is significant.

Table 1 : Separation ability analysis among different vehicle categories before color correction.

Colors	Red	Green	Blue	Yello	Silve	Black	W
Red	X	0.041	0.02	0.031	0.038	0.010	0.01
Green	0.04	X	0.03	0.019	0.041	0.061	0.05
Blue	0.02	0.032	X	0.014	0.031	0.020	0.02
Yell	0.03	0.019	0.01	X	0.030	0.038	0.02
Silver	0.03	0.041	0.03	0.030	X	0.032	0.03
Black	0.01	0.061	0.02	0.038	0.032	X	0.03
W	0.01	0.050	0.02	0.028	0.034	0.035	X

Table 2 : Separation ability analysis among different vehicle categories after color correction.

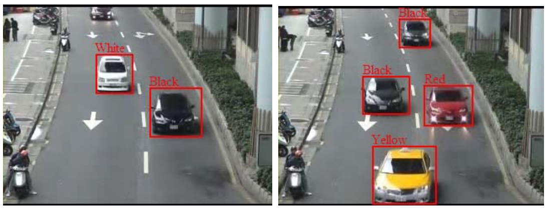
	Red	Green	Blue	Yell	Sil	Black	W
Red	X	0.063	0.060	0.177	0.091	0.021	0.032
Green	0.06	X	0.03	0.128	0.121	0.065	0.043
Blue	0.06	0.032	X	0.052	0.059	0.060	0.034
Yell	0.17	0.128	0.05	X	0.73	0.160	0.167
Silver	0.091	0.121	0.059	0.73	X	0.131	0.126
Black	0.021	0.065	0.060	0.160	0.131	X	0.091
W	0.032	0.043	0.034	0.167	0.126	0.091	X

Table 3: Confusion matrix of vehicle classification before correction.

Colors	B	S	W	Y	Red	Green	Blue
Black	83.95	5.89	0.00	0.00	0.29	3.98	5.89
Silver	7.49	75.03	13.61	0.00	0.00	1.37	2.50
White	0.00	13.4	84.98	0.96	0.64	0.00	0.00
Yellow	1.34	0.67	0.00	97.1	0.67	0.22	0.00
Red	1.24	0.50	0.00	0.50	97.28	0.25	0.25
Green	8.39	4.73	0.00	0.00	0.00	84.73	0.22
Blue	10.65	1.90	0.00	0.00	0.00	4.56	82.89

Table 4: Confusion matrix of vehicle classification after correction.

Color	B	S	W	Y	Red	G	Blue
Black	92.74	5.28	0.00	0.00	0.00	1.15	0.83
Silver	4.88	85.99	7.65	0.00	0.00	0.72	0.76
White	3.36	3.70	92.94	0.00	0.00	0.00	0.00
Yellow	0.58	0.75	0.29	98.38	0.00	0.00	0.0
Red	1.58	0.27	0.00	0.00	98.14	0.00	0.0
Green	5.21	2.17	0.00	0.00	0.00	92.57	0.04
Blue	6.34	0.99	0.00	0.00	0.00	2.70	89.97



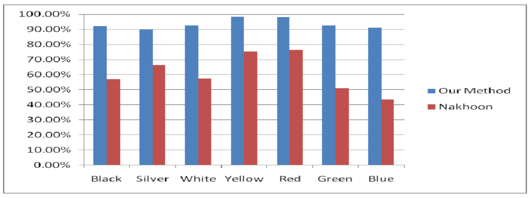
(a) (b)

Fig. 9 : Results of classification when the road included multiple vehicles.

In addition to the separation ability analysis, we also evaluate the performance improvement of vehicle classification before and after color correction using real data. A testing database included 16648 vehicles was adopted, where 3243 images were used for white vehicles, 2783 ones for black vehicles, 2234 ones for silver vehicles, 1832 ones for red vehicles, 2303 ones for green vehicles, 2523 ones for blue vehicles, and 1730 ones for yellow vehicles. Table 3 and Table 4 list the confusion matrices of vehicle classification before/after correction. Compared Table 4 with Table 3, significant accuracy improvements of vehicle classification can be found. Fig. 9 shows the results of vehicle classification when the road included multiple vehicles.

For comparisons, another SVM-based method [12] was also implemented. This method cannot well handle the problem of lighting changes. Table 5 indicates the comparisons between our proposed method and [12]. The average accuracy of [12] is 61.04%. The average accuracy of our method is 94.02%. Clearly, our method performs much better than [12] in vehicle classification.

Table 5: Comparison results between our proposed method and [12].



REFERENCES

[1] R. Cucchiara, M. Piccardi, and P. Mello, “Image analysis and rule-based reasoning for a traffic monitoring system,” *IEEE Transactions on ITS*, vol. 1, no. 2, pp. 119-130, 2002.

[2] O. Javed, K. Shafique, and M. Shah, “Appearance modeling for tracking in multiple non-overlapping cameras,” IEEE Conf. on CVPR, vol. 2, pp.26-33, 2005.

[3] Y. Sheikh, X. Li, and M. Shah, “Trajectory association across non-overlapping moving cameras in planar scenes,” IEEE Conf. on CVPR, pp. 1-7, Jun. 2007.

[4] T.-H. Chang and S.-G. Gong, “Tracking Multiple People with a Multi-Camera System,” IEEE Workshop on Multi-Object Tracking, pp.19-26, 2001.

[5] S. M. Khan and M. Shah, “A multi-view approach to tracking people in crowded scenes using a planar homography constraint,” European Conference of Computer Vision, vol. 4, pp.133-146, 2006.

[6] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Spectral sharpening: Sensor Transformations for Improved Color Constancy,” The Optical Society of America, vol. 11, no. 5, pp.1553-1564,May 1994.

[7] T. Horprasert, *et al.* “A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection,” IEEE ICCV, pp.1-19, 1999.

[8] C.-C. Chang and C.-J. Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] E. Reinhard, *et al.*, “Color transfer between images,” IEEE Computer Graphics and Applications, vol. 21, no. 5, pp.34-41, 2001.

[10] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, John Wiley & Sons, Inc., New York, 2001.

[11] Y. Xiang, B. Zou, and H. Li, “Selective color transfer with multi-source images,” *Pattern Recognition Letters*, vol. 30, no. 7,pp. 682-689, May 2009.

[12] N. Baek, et al., “Vehicle color classification based on the support vector machine method,” *Communications in Computer and Information Science*, vol. 2, Part 24, pp. 1133-1139, 2007.