

# A view-invariant and anti-reflection algorithm for car body extraction and color classification

Hui-Zhen Gu · Suh-Yin Lee

Published online: 22 February 2012  
© Springer Science+Business Media, LLC 2012

**Abstract** This study proposes an intelligent algorithm with tri-state architecture for real-time car body extraction and color classification. The algorithm is capable of managing both the difficulties of viewpoint and light reflection. Because the influence of light reflection is significantly different on bright, dark, and colored cars, three different strategies are designed for various color categories to acquire a more intact car body. A SARM (Separating and Re-Merging) algorithm is proposed to separate the car body and the background, and recover the entire car body more completely. A robust selection algorithm is also performed to determine the correct color category and car body. Then, the color type of the vehicle is decided only by the pixels in the extracted car body. The experimental results show that the tri-state method can extract almost 90% of car body pixels from a car image. Over 98% of car images are distinguished correctly in their categories, and the average accuracy of the 10-color-type classification is higher than 93%. Furthermore, the computation load of the proposed method is light; therefore it is applicable for real-time systems.

**Keywords** View-invariant · Light reflection · Tri-state architecture · Image segmentation · Car body extraction · Color classification

## 1 Introduction

With the rapid growth of surveillance equipments, recognizing vehicles by their visual features is currently a popular issue. Color is a dominant visual feature on vehicles, and is widely used in intelligent transportation systems (ITS) and content-based image retrieval (CBIR) systems. For ITS, the color feature is used to enhance car recognition accuracy for

---

H.-Z. Gu (✉) · S.-Y. Lee  
Department of Computer Science and Information Engineering, National Chiao Tung University,  
1001 Ta Hsueh Rd., Hsinchu 300 Taiwan, Republic of China  
e-mail: hcku@cs.nctu.edu.tw

S.-Y. Lee  
e-mail: sylee@cs.nctu.edu.tw

crime detection, entrance security, or road charging. For CBIR systems, vehicle colors assist users to query their desired vehicle images from the Internet or a large image database.

Current vehicle color classification methods construct color histograms from car images before matching the color distribution with the templates. Kim et al. [14] determined the optimal bin numbers of the histogram of hue, saturation and intensity axes to acquire optimal performance. In [16], a method based on vector matching was proposed and the color distribution in HSI color space was claimed to be more distinguishable than in the RGB or YCbCr color space. In [3], an example-based algorithm was proposed, which exhibited the potential to reduce the effects caused by lighting variations when the vehicle pose was restricted.

The Support Vector Machine (SVM) is a useful tool for classification when combined with the color histogram methods. In [7], the SVM method was introduced to classify the dominant color of vehicles. In [2], not only color, but the sizes of cars were classified by SVM. In [21], a grid kernel method, which detects color information from not only a pixel but also its neighboring pixels, was proposed to achieve a superior recognition rate. However, these SVM related techniques usually encounter a considerable problem. A number of undesired pixels on the windshield, headlights or the background often interfere with the color histogram, thereby decreasing the classification accuracy. Therefore, recent studies only considered the pixels that were included in the car body to calculate color histograms.

A car body comprises metallic parts, as follows: the hood, the roof, and the side panels (excluding the windshield, headlights, and wheels). The work of extracting the car body from a car image is another challenging problem and usually requires a number of restrictions. In [4], the vehicle was captured in the front view. Subsequently, a fixed position rectangle, which was guaranteed to cover the most significant part of the car body, such as the hood of the car, was acquired. A homogeneous region with a similar color was extracted within the rectangle and the vehicle color was identified by only the pixels of this homogeneous region. In [19], cars were captured in the rear view. The significant part of the car body was also extracted and the vehicle color was identified by the pixels within the significant part.

The restriction, in that the camera viewpoint is limited to the front or rear views, must be removed for wider and more realistic applications. In [26], the camera was fixed and the lane direction was predetermined. A rectangle box, in which the region of hood can be detected, was subsequently defined and termed as the first sight window. The frame difference technique [27] was used to localize the vehicle within the window. A homogeneous region on the vehicle body was subsequently extracted for color classification. In [25], the vehicle was assumed to be located near the center of the tested image and occupied most parts of the image. A number of removal rules were defined to reject the undesirable pixels. The remaining parts were considered the car body. The background subtraction technique [10] was adopted in this approach to generate an image that enclosed the vehicle. For a moving vehicle or known background cases, as in ITS scenarios, the frame difference method [27] or the background subtraction method [10] is able to detect the contour of the car; therefore, the tested car can be placed near the center of the image. However, if the tested car is static and the background is unknown, as in CBIR systems, the car images may be captured from any viewpoint and appear in any location. It is more difficult to determine the correct car body in these scenarios.

The image segmentation techniques may be used to acquire the car body on a vehicle image without any viewpoint and location constraints. Each image is segmented into multiple regions, and the car body is composed of one or several regions. Jain et al. [13]

provided a review of data clustering and image segmentation techniques. The boundary-based and color-based approaches are commonly used in image segmentation. The boundary-based approach, such as seeded region growing [20], selected abundant disconnected pixels as the seeds. Each seed generates a region composing of the neighboring pixels and contains similar colors of the seed. In the color-based approaches, [1] (and its previous paper [18]) selected a set of hills (local maximum) in the color histogram, and the pixels were grouped into the hills based on the minimal color difference. If the bin number of the histograms is determined optimally, all pixels that belong to one significant object may be grouped into one region. The significant object is subsequently extracted and the color of the object is classified.

The boundary-based approach manages several objects on an image. However, a number of notable regions may be overlooked if the seeded region growing process encounters obstacles, such as a group of pixels whose colors are considerably different to the neighboring pixels. Therefore, when the approach is applied in the car body extraction topic, a number of significant parts of the car body may be easily overlooked, because the intensity of the car body varies considerably due to non-homogeneous light reflection. Hence, the color-based approach is adopted in the proposed system. Since the above approaches do not seriously consider the various effects of the non-homogeneous light reflection on different color cars, they are usually bothered with the effect of the non-homogeneous light.

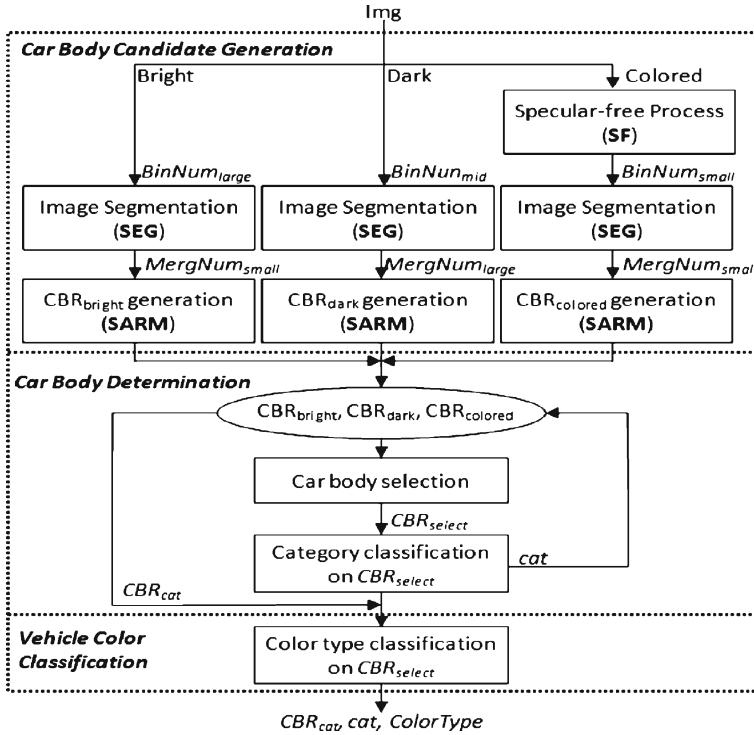
In [23], a specular-free processing was proposed to effectively overcome the non-homogeneous light reflection for general colored objects. Hence, the specular-free processing technique was integrated in our system to deal with the non-homogeneous light reflection for colored cars. However, the main side effect of the technique is that the extracted car bodies and classified colors for grayscale cars are destroyed.

This study proposes a tri-state architecture system that properly processes the tested cars in colored, bright, and dark categories. The color-based approach [1] is adopted to segment the image first. Subsequently, we propose a SARM (Separating and Re-Merging) algorithm to separate the car body and the background more completely. In addition, the algorithm carefully merges the regions that belonged to the car body depending on the color category. Because the effects of the non-homogeneous light reflection vary considerably in the three color categories, different strategies are designed and the critical parameters for each category are individually selected. The specular-free processing [23] is also used before the segmentation processing for colored cars. The proposed system is capable of managing both the difficulties of various viewpoints and non-homogeneous light reflection, thereby facilitating extraction of the car body, and enabling accurate classification of vehicle colors.

The remainder of this paper is organized as follows. Section 2 presents an overview of the proposed system. In Section 3, three car body candidates are generated by using various strategies, for colored, bright and dark categories. Section 4 provides the criteria for selecting the car body from the candidates and the adopted color classification mechanism is also introduced in this section. The experimental results and conclusions are offered in Sections 5 and 6, respectively.

## 2 System overview

Figure 1 shows the flowchart of the proposed car body extraction and color classification system. The terms utilized in this figure are explained in Table 1. The input of the system is a



**Fig. 1** The framework of the car body extraction and color classification mechanism

car image ( $Img$ ) of any color type. The system extracts the car body region ( $CBR_{cat}$ ) and distinguishes the color category ( $cat \in \{colored, bright, dark\}$ ) and the color type ( $ColorType \in \{white, gray, black, red, orange, yellow, green, purple, blue, pink\}$ ) of the car. The system comprises three major steps: car body candidate generation, car body determination, and vehicle color classification.

By observing numerous real vehicle images, we found that the influence of light reflection differs in colored, bright and dark vehicles. For example, under light reflection, RGB values of colored and dark cars would alter much more than those of bright cars. In

**Table 1** Terminology table

Terms	Definition
$Img$	an input vehicle image
$BinNum_{\{large, mid, small\}}$	predefined small, medium and large bin numbers of histogram
$MergNum_{\{large, small\}}$	predefined small and large merge times
$CBR_{\{bright, dark, colored\}}$	car body candidates for bright, dark or colored categories
$CBR_{select}$	the selected car body among the candidates
$cat$	the classified category on $CBR_{select}$
$CBR_{cat}$	the final extracted car body based on current category
$ColorType$	the final classified color type of the input vehicle

addition, because light reflection generally originates from strong bright luminance, light reflection for colored cars can be distinguished more easily from the chromaticity of the car body. Therefore, this study proposes tri-state architecture for extracting an intact car body under different light situations. Three car body candidates are generated simultaneously for the three color categories. The tested car image is optimally processed for each color category, and finally, one of the candidates is determined as the extracted car body.

To generate the three candidates, an image segmentation technique [1] based on color histogram is employed first to segment the input image into multiple color coherent regions. Generally, the number of the segmented regions is proportional to the bin number ( $BinNum$ ) of color histogram. If the bin number is too small, a number of regions in the background are easily included in the extracted car body. However, when the bin number is too large, the car body region is split into numerous small pieces, and some of them may be excluded from the extracted car body. To acquire preferable segmentation, different bin numbers should be given for different categories.

For colored cars, because the effect of light influence is variable, a specular-free processing is adopted before segmenting the image. The specular-free process [23] uses the chromaticity information to generate a specular-free image. Subsequently, the specular-free image is segmented by adopting a small number of color bins ( $BinNum_{small}$ ) and a segmented image is produced. For bright and dark cars, since no chromaticity information exists in the car body, the specular-free processing is not applied to distinguish the intensity of the car body. Large and medium bin numbers ( $BinNum_{large}$ ,  $BinNum_{mid}$ ) are required for bright cars and dark cars individually, to prevent the car body from grouping with the background. However, more small pieces are generated when larger bin numbers are adopted. Therefore, a merging process is necessary to recover the whole car body.

To generate the car body candidates ( $CBR_{bright}$ ,  $CBR_{dark}$ , and  $CBR_{colored}$ ) more accurately, a SARM (Separating and Re-Merging) algorithm is proposed. The algorithm considers not only the color feature but also the spatial relationships between each pairs of segmented regions on vehicle images. Since the influences of light reflection on bright, dark and colored cars are various, different values are assigned to the number of merging times. For dark cars, a large merging number ( $MergNum_{large}$ ) is given, since more small pieces are produced after segmentation. For colored and bright cars, the influence of light reflection will not make so many small pieces as in the dark car cases. Thus, a small merging number ( $MergNum_{small}$ ) is sufficient.

When all car body candidates:  $CBR_{bright}$ ,  $CBR_{dark}$ ,  $CBR_{colored}$ , for the three categories have been prepared, a selection scheme that compares the significance and compactness of each candidate is proposed. After processing this scheme, a car body ( $CBR_{select}$ ) is selected from the candidates and the color category of the selected car body is classified by SVM technique. Based on the classified category, the candidate ( $CBR_{cat}$ ) which has identical category to the classified result is obtained as the extracted car body.

After the car body of the input car has been extracted, the color type ( $ColorType$ ) of the vehicle is decided only according to the pixels in the extracted car body. A hierarchically structured SVM is adopted in this paper. In the first layer, a binary SVM is utilized to classify the vehicle into colored or grayscale class. If the vehicle belongs to the grayscale class in the first layer, a multi-class SVM is designed to classify the image into a black, gray, or white car in the second layer. If the vehicle is identified to belong to the colored category in the first

layer, another multi-class SVM is designed to classify the image into a red, orange, yellow, green, blue, purple or pink car in the second layer.

### 3 Car body candidates generation

Three different strategies are designed for various color categories because the influence of light reflection varies considerably on bright, dark, and colored cars. Section 3.1 displays the implementation of the hill-climbing algorithm to segment the tested car image. Section 3.2 shows specular-free processing prior to image segmentation to alleviate the reflection effect for colored cars. The SARM (Separate and Re-Merging) algorithm which can highly improve the capability of separating the car body and the background and recovering the entire car body is proposed in Section 3.3. The process to generate triple car body candidates for dark, bright, and colored cars individually is described in Section 3.4. Process justification and parameter selection are performed in Section 3.5.

#### 3.1 Image segmentation

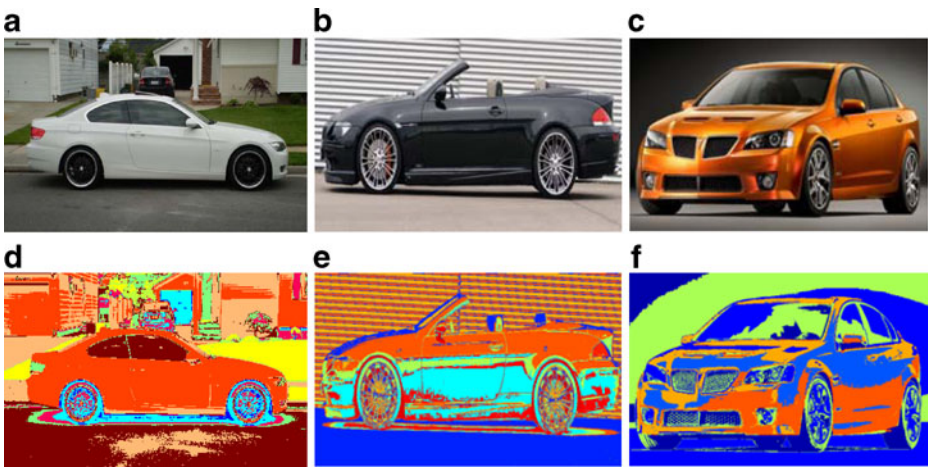
To segment a vehicle image into multiple regions with coherent color attributes, abundant image segmentation techniques have been proposed. Among these studies, a color-based image segmentation algorithm: hill climbing [1, 18] is adopted in the proposed system, due to its sufficient effectiveness and efficiency for application in real-time transportation systems.

In our system, the color histogram in the Lab color model is computed, and a set of local peaks on the histogram are searched. The adjacency pixels sharing similar colors to the color of the peak are grouped together and considered a color coherent segment. The number of peaks calculated via hill-climbing algorithm is under control by the number of histogram bins; hence, influencing the number of segments (*SegNum*) is possible by adjusting the bin number (*BinNum*). After image segmentation, a segmented image (*SImg*) is generated and composed of multiple regions ( $Reg^1, \dots, Reg^{SegNum}$ ), as Eq. 1.

$$SImg = \{Reg^i | 1 \leq i \leq SegNum\} = SEG(Img, BinNum) \quad (1)$$

Figure 2(a)–(c) show a white, a black, and an orange car images. The segmented results via the hill-climbing algorithm are shown in Fig. 2(d)–(f). The pixels grouped together in the color histogram are represented by a unique color. The color coding scheme is based on the number of regions and the built-in MATLAB colormap “jet” [8]. If the number of region is *SegNum*, *SegNum* colors selected from the colormap are acquired to draw the image.

From Fig. 2(d)–(f), some problems can be found. First, some background pixels are misclassified into the car body. Second, the car body may be disrupted into multiple large partitions. Third, some small fragments appear on the car body due to light reflection and these fragments have extremely different colors to the car body. To overcome these phenomena, the specular-free processing and the proposed SARM algorithm are addressed in Sections 3.2 and 3.3, respectively, to segment the car body from images.



**Fig. 2** **a** a white car image, **b** a black car image, **c** an orange car image. The image segmentation results by hill-climbing of the **d** white, **e** black, and **f** orange car image

### 3.2 Specular-free processing

The colored cars can be processed via a specular-free operator to alleviate the influence of light reflection. The luminance reflected from an object (including cars) can be formulated by a linear combination of diffuse and specular reflections:

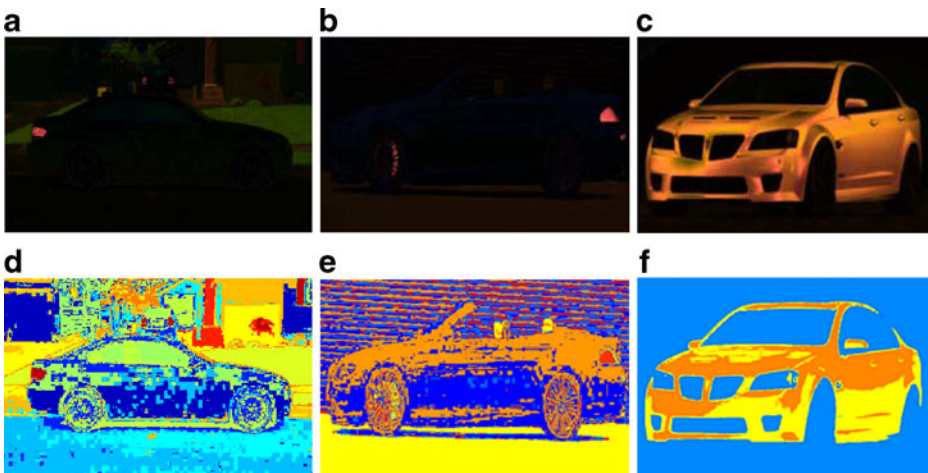
$$Img_c(x) = m_d(x) \times \Lambda_c(x) + m_s(x) \times \Gamma_c \quad (2)$$

where  $Img_c(x)$ ,  $m_d(x)$ ,  $m_s(x)$ ,  $\Lambda_c(x)$ , and  $\Gamma_c$  are the intensity value in RGB color space (i.e.  $c \in \{r, g, b\}$ ), the diffuse coefficient, the specular coefficient, the diffuse chromaticity, and specular chromaticity of a pixel  $x$  on an image  $Img$ .

The diffuse reflection term  $m_d(x) \times \Lambda_c(x)$  reveals the actual color of the object, while the specular reflection term  $m_s(x) \times \Gamma_c$  originates from the undesirable light reflection. Numerous studies [11, 17, 23] have developed methods for separating the diffuse and specular reflections. This study adopts a mechanism [23] capable of transforming the specular image into the specular-free image (i.e.  $SF_c(x) = m_d(x) \times \Lambda_c(x)$ ) to overcome the influence of light reflection in the car body extraction problem.

To construct a specular-free image, a luminance-normalized image with pure white specular components is first produced. A non-linear shift function is then computed to transform the intensity and chromaticity of each pixel to the normalized image while retaining the color information. Figure 3(a)–(c) show the specular-free images derived from the images in Fig. 2(a)–(c).

In Fig. 3(a) and (b), specular-free processing is unsuitable for application on grayscale (dark or bright) cars. Since the diffuse components of all grayscale pixels approximate to zero, the car body is severely destructed and becomes ever more difficult for segmentation. Hence, this study only applies specular-free processing for colored cars. In Fig. 3(c), a specular-free processing is performed on the orange car. The segmented result of Fig. 3(c) is shown in Fig. 3(f). Since the influence of light



**Fig. 3** a–c The specular-free images of the white, black, and orange cars, d–f the segmented results of the specular-free images via hill-climbing algorithm

reflection has been evidently alleviated, the car body of the orange car (and also for other colored cars) can be segmented more effectively. The segmented car images are processed via the separating and re-merging method to improve segmentation performance, as described in Section 3.3.

### 3.3 Separating and re-merging

From the previous sections, the significant part of the vehicle image may include impure background pixels (environment) or improper foreground pixels (windshield, lights, wheels, etc.); therefore the segmented regions must be “separated”. Because the car body may be disrupted into several large partitions or small fragments, a “merging” process must be performed next to recover the entire car body. In addition, the large partitions and the small fragments usually have various characteristics. For example, a small fragment incurred by the strong light reflection generally has considerably high intensity. Whereas, a large partition caused by environmental luminance usually has the intensity value close to that of the car body. Therefore, the merging strategies of small fragments and large partitions must be designed differently.

Three functions for large partition separating, small fragment merging, and large partition merging are proposed, which are referred to *partitionSeparate*, *microMerge*, and *macroMerge*, respectively. These functions are integrated into the SARM (Separating and Re-Merging) algorithm. To separate the background and merge multiple partitions into a car body, the algorithm simultaneously considers color feature and spatial relationships and properly controls the parameters. The procedures of these functions are described as follows.

**Function *partitionSeparate*** This function separates the disconnected partitions of each consistent color region in the original segmented image (*SImg*) produced by the hill-



climbing algorithm. Because the algorithm segments images only considering the color features, the pixels with a similar color are clustered regardless of whether they are far away from each other. This function considers the spatial relationships between the regions to generate a new segmented image (*SepImg*). It contains three steps as shown in List 1.

In step 1, the regions in the input segmented image, the sizes of which are larger than a size-threshold ( $th_{size}=200$  pixels, empirically), are collected; otherwise, the small regions are not separated. In step 2, for each collected region, the 8-connectivity is accessed by the algorithm (*8-ConnectivityCheck*) in [12]. After the process of *8-ConnectivityCheck*, each region ( $Reg^i$ ) is separated into  $k$  partitions ( $Par_1^i, \dots, Par_k^i$ ). In step 3, each partition is considered as a new region, and assigned a distinct label. Finally, the new labeled regions ( $SepReg^1, \dots, SepReg^m$ ) compose of a new segmented image (*SepImg*).

<p><b>Procedure:</b> [<i>SepImg</i>] = <i>partitionSeparate</i> (<i>SImg</i>, <math>th_{size}</math>).</p> <p><b>Input:</b> <math>SImg = \{Reg^1, \dots, Reg^n\}</math>: a segmented image created by hill-climbing and composed of a set of segmented regions; the number of regions is <math>n</math>  <math>th_{size}</math>: a size threshold to determine large and small regions</p> <p><b>Output:</b> <math>SepImg = \{SepReg^1, \dots, SepReg^m\}</math>: a new segmented image composed of more segmented regions; the number of regions is <math>m</math> (<math>m \geq n</math>)</p> <p><b>Step</b></p> <p>0     <math>m = 0</math>;   // initialization of the number of new segmented regions</p> <p>      <b>For</b> <math>1 \leq i \leq n</math></p> <p>1         <b>If</b> <math>size(Reg^i) &gt; th_{size}</math>                // Separating the region into <math>k</math> partitions by checking 8 neighboring                // pixels of each pixel in the region.</p> <p>2             <math>[Par_1^i, \dots, Par_k^i] = 8\text{-ConnectivityCheck}(Reg^i)</math>;</p> <p>3             <b>For</b> <math>1 \leq j \leq k</math>                    <math>SepReg^{m+j} = Par_j^i</math>;</p> <p>              <b>End</b>                    <math>m = m + k</math>;</p> <p>              <b>Else</b> <math>m = m + 1</math>;   <math>SepReg^m = Reg^i</math>;</p> <p>              <b>End</b></p> <p>      <b>End</b></p> <p>      <b>End</b></p>
--

List 1 The procedure of *partitionSeparate* function

**Function *microMerge*** This function merges the small size regions on the input segmented image (*SepImg*) and creates a new segmented image (*MergSepImg*) only containing sufficiently large regions. The function contains three main steps, which are described in List 2.

In step 1, the segmented regions are classified into small-size fragments ( $smallFrag^1, \dots, smallFrag^{smallCount}$ ) and large-size partitions ( $MergSepReg^1, \dots, MergSepReg^{largeCount}$ ) by comparing their areas with  $th_{size}$ .

In step 2, for each pair of small-size fragment ( $smallFrag^i$ ) and large-size partition ( $MergSepReg^j$ ), check whether the large-size partition is neighboring to the small-size fragment. If any pixel of the large-size partition is in the neighboring area ( $7 \times 7$  pixels) of the boundary pixels of the small-size fragment, they are considered to be neighbored. The

function NeighborTest will return 1, and then the step 3 is executed. Otherwise, the function NeighborTest returns 0, and then the step 3 is skipped.

In step 3, the hue, saturation, and intensity differences ( $diff_{hue}$ ,  $diff_{sat}$ ,  $diff_{int}$ ) between the small fragment and all its neighboring large partitions are computed. Among the large partitions whose color differences ( $diff_{hue}$ ,  $diff_{sat}$ ,  $diff_{int}$ ) are all smaller than the corresponding thresholds ( $th_{hue}$ ,  $th_{sat}$ ,  $th_{int}$ ), the largest partition is selected. The small fragment is merged into the selected large partition. Step 2 and step 3 repeat until all pairs of small-size fragments and large-size partitions have been checked.

The thresholds differ on categories because the light reflection influence on dark cars differs from that of the white or colored cars. The thresholds for each category are discussed in Section 3.4. Finally, the large merged partitions compose the new segmented image ( $MergSepImg$ ). Although the fragments are merged into the large partitions, their colors are not counted to the average color of the large partitions. The colors of the merged fragments are adapted to be equal to the average color of the large partitions which the fragments are merged to.

```

Procedure: [ $MergSepImg$ ] = microMerge( $SepImg$ ,  $th_{color}$ ).
Input:  $SepImg = \{SepReg^1, \dots, SepReg^m\}$ : the input segmented image and its regions
         $th_{color} = [th_{hue}, th_{sat}, th_{int}]$ : difference thresholds of hue, saturation, and intensity
Output:  $MergSepImg = \{MergSepReg^1, \dots, MergSepReg^{largeCount}\}$ : the segmented image after
        merging small regions into large ones; the regions number is  $largeCount$ 

Step
1  For  $1 \leq i \leq m$ 
    If ( $size(SepReg^i) < th_{size}$ )  $smallCount++$ ;  $smallFrag^{smallCount} = SepReg^i$ ;
    Else  $largeCount++$ ;  $MergSepReg^{largeCount} = SepReg^i$ ;
    End
  End
2  For  $1 \leq i \leq smallCount$ 
     $max\_size = 0$ ;
    For  $1 \leq j \leq largeCount$ 
      // determine the small fragment  $smallFrag^i$  and the large partition  $MergSepReg^j$ 
      // whether they are appearing in the neighboring ( $7 \times 7$ ) area or not
      If NeighborTest( $smallFrag^i$ ,  $MergSepReg^j$ ) == 1
3    [ $diff_{hue}$ ,  $diff_{sat}$ ,  $diff_{int}$ ] = ColorDiff (avg ( $smallFrag^i$ ), avg( $MergSepReg^j$ ));
      If ( $diff_{hue} < th_{hue}$  &  $diff_{sat} < th_{sat}$  &  $diff_{int} < th_{int}$ ) & ( $size(MergSepReg^j) > max\_size$ )
         $max\_size = size(MergSepReg^j)$ ;
         $max\_index = j$ ;
      End
    End
     $MergSepReg^{max\_index} = MergSepReg^{max\_index} \cup smallFrag^i$ ;
  End

```

List 2 The procedure of *microMerge* function

**Function *macroMerge*** This function selects one or multiple regions from the new segmented image ( $MergSepImg$ ) to compose the car body ( $CBR_{cat}$ ). The function consists of four steps and is described in List 3.

**Procedure:**  $[CBR_{cat}] = \text{macroMerge}(\text{MergSepImg}, th_{color}, \text{MergNum}, cat)$ .

**Input:**  $\text{MergSepImg} = \{ \text{MergSepReg}^1, \dots, \text{MergSepReg}^{\text{largeCount}} \}$ : the input segmented image  
 $th_{color} = (th_{hue}, th_{sat}, th_{int})$ : the color difference threshold  
 $\text{MergNum}$ : the number of iteration for merging  
 $cat$ : the current category

**Output:**  $CBR_{cat} = \{ CBR_{cat}^i, \dots, CBR_{cat}^k \}$ : one or multiple regions selected as the car body

**Step**

- 1  $\text{ConnectTable} = \text{ConstructConnectTable}(\text{MergSepImg});$
- 2 **For**  $1 \leq i \leq \text{largeCount}$ 
  - $[\text{InnerImg}, \text{OuterImg}] = \text{DivideInnerOuter}(\text{MergSepReg}^i);$
  - $\text{InPxNum}[i] = \# \text{ pixels} \in \text{MergSepReg}^i \cap \text{InnerImg};$
  - $\text{OutPxNum}[i] = \# \text{ pixels} \in \text{MergSepReg}^i \cap \text{OuterImg};$
  - If**  $(\text{OutPxNum}[i] > th_{\text{OutPxNum}})$   $\text{InPxNum}[i] = 0;$  **End**
  - If**  $(\text{OutPxNum}[i] / \text{InPxNum}[i] > th_{\text{OutInRatio}})$   $\text{InPxNum}[i] = 0;$  **End**
  - If**  $(cat == \text{'bright'} \ \& \ \text{avgSat}(\text{MergSepReg}^i) > th_{\text{upbound}_{sat}}^{\text{upbound}})$   $\text{InPxNum}[i] = 0;$
  - Elseif**  $(cat == \text{'dark'} \ \& \ \text{avgSat}(\text{MergSepReg}^i) > th_{\text{upbound}_{sat}}^{\text{upbound}})$   $\text{InPxNum}[i] = 0;$
  - Elseif**  $(cat == \text{'colored'} \ \& \ \text{avgSat}(\text{MergSepReg}^i) < th_{\text{lowbound}_{sat}}^{\text{lowbound}})$   $\text{InPxNum}[i] = 0;$
  - End**
  - End**
  - If**  $cat == \text{'bright'}$ 
    - $\text{SeedCandidates} = \text{TopThreeMax}(\text{InPxNum});$
    - $seed = \text{MaxIntensity}(\text{SeedCandidates});$
  - Elseif**  $cat == \text{'dark'}$ 
    - $\text{SeedCandidates} = \text{TopThreeMax}(\text{InPxNum});$
    - $seed = \text{MinIntensity}(\text{SeedCandidates});$
  - Else**  $//cat == \text{'colored'}$ 
    - $seed = \text{TopOneMax}(\text{InPxNum});$
  - End**
  - $\text{InPxNum}[\text{index}(seed)] = 0;$   $// \text{ clear inner pixel number of the seed region}$
- 3  $iter\_count = 0;$   $\text{MacroMergList} = \text{Null};$   $list\_num = 0;$ 
  - Do**  $\text{maxPartIdx} = \text{maxIndex}(\text{InPxNum});$   $// \text{ get the index of max region in InPxNum}$
  - $iter\_count ++;$
  - $[\text{diff}_{hue}, \text{diff}_{sat}, \text{diff}_{int}] = \text{ColorDiff}(\text{avg}(seed), \text{avg}(\text{MergSepReg}^{\text{maxPartIdx}}));$
  - If**  $\text{diff}_{hue} < th_{hue} \ \& \ \text{diff}_{sat} < th_{sat} \ \& \ \text{diff}_{int} < th_{int}$ 
    - If**  $\text{ConnectTable}[seed, \text{MergSepReg}^{\text{maxPartIdx}}] == 1$ 
      - $CBR_{cat} = seed \cup \text{MergSepReg}^{\text{maxPartIdx}};$
      - For**  $1 \leq i \leq list\_num$ 
        - If**  $\text{ConnectTable}[\text{MacroMergList}^i, \text{MergSepReg}^{\text{maxPartIdx}}] == 1$ 
          - $CBR_{cat} = seed \cup \text{MacroMergList}^i;$
          - $\text{MacroMergList} = \text{MacroMergList} - \text{MacroMergList}^i;$
        - End**
      - End**
    - Else**  $list\_num ++;$ 
      - $\text{MacroMergList} = \text{MacroMergList} \cup \text{MergSepReg}^{\text{maxPartIdx}};$
    - End**
    - $\text{InPxNum}[\text{maxPartIdx}] = 0;$
  - 4 **While**  $iter\_count < \text{MergNum} \ \& \ \text{size}(\text{MergSepImg}^{\text{maxPartIdx}}) > 0$

List 3 The procedure of *macroMerge* function

In step 1, a connection table is constructed which describes the connectivity of each pair of the large partitions. For each boundary pixel in each large partition, check the other large partitions whether appearing in the neighboring ( $7 \times 7$  pixels) area. If two large partitions have over  $num_{neighbor}$  pixels (empirically,  $num_{neighbor}=60$ ) appearing in the neighboring area, these two partitions are considered to be connected. The connectivity relationships between all pairs of the partitions in *MergSepImg* are recorded in the connection table.

In step 2, a seed region that will merge the other regions is selected. In general, the central part of an image is more significant than the boundary part [15]. Therefore, the image is firstly divided into the following two parts: inner part (*InnerImg*), and outer part (*OuterImg*) by two parameters  $\delta_x, \delta_y$ . The parameters are defined as a fraction (for example, 1/10) of the width and the height of the image. The area with width ranging from  $[\delta_x, wid-\delta_x]$  and height ranging from  $[\delta_y, len-\delta_y]$  is considered as the inner part, and the remaining area is considered as the outer part. For each segmented region, the pixel number in the inner part (*InPxNum*) and the pixel number in the outer part (*OutPxNum*) are both computed. If a region which its pixels in the outer part are more than a predefined threshold ( $th_{OutPxNum}$ ) or its ratio of *OutPxNum* over *InPxNum* is greater than another predefined threshold ( $th_{OutInRatio}$ ), the region is considered as the background. Subsequently, the number of the pixels in the inner part is ignored and set to zero. Then, this region will not be selected as the seed region.

Furthermore, the average saturation of each region is computed and used to decide the seed region. If the input category (*cat*) is bright or dark, the regions with high saturation (i.e. larger than a predefined threshold  $th_{sat}^{upbound}$ ) are not selected as the seed region and their pixel numbers in the inner part are set to zero. If the input category is colored, the regions with low saturation (i.e. lower than a predefined threshold  $th_{sat}^{lowbound}$ ) are not selected as the seed region and their pixel numbers in the inner part are set to zero. Next, the top three largest partitions in *MergSepImg* and passed the above tests are selected as the seed candidates. If the category is bright, the candidate which has the highest average intensity is selected as the seed region. If the category is dark, the candidate which has the lowest average intensity is selected as the seed region. If the category is colored, the candidate with the largest area is selected as the seed region.

In step 3, the partition (*MergSepReg*<sup>maxPartIdx</sup>) with maximal area is selected and excluded at the end of loop. The connectivity and average color of the selected partition is assessed with the seed region. If the color difference between the partition and the seed is smaller than the thresholds ( $th_{hue}, th_{sat}, th_{int}$ ), the connectivity between them are further assessed. The partition is merged into the seed region if it is connected to the seed. Otherwise, it is added into a merge list (*MacroMergList*). All of the partitions in the merge list are assessed again when any new partition is merged into the seed region. If the connectivity between any partition in the merge list and the new partition (*MergSepReg*<sup>maxPartIdx</sup>) exists, the partition in the merge list is excluded from the merge list and merged into the seed region. In step 4, the number of iteration (*iter\_count*) and the input parameter *MergNum* are assessed. If they are equal, the function returns the merged result. Otherwise, step 3 is repeated.

The car body of the tested car can be recovered more completely through the processing of the SARM algorithm. Figure 4 shows the segmentation results after the SARM algorithm processing. A number of weaknesses, as shown in Figs. 2 and 3, are effectively overcome.

### 3.4 Tri-state car body candidate generation

Three car body candidates for dark, bright and colored cars are generated. The procedure is formed with the image segmentation technique, the specular-free operator, and the SARM algorithm, as described in Sections 3.1, 3.2, and 3.3, respectively. Various input parameters are assigned to extract the intact car body in any category, and the methodology is described in List 4.



**Fig. 4** a–c The extracted car body candidates of the a white, b black, and c orange cars after the process of the SARM algorithm

In List 4, the input data comprised the following: (1) a vehicle image ( $Img$ ), (2) the category of current state:  $cat \in \{bright, dark, colored\}$ , (3) the bin number of color histogram ( $BinNum$ ) for each category, (4) the merging times ( $MergNum$ ) of **macroMerge** for each category, and (5) several predefined thresholds ( $th_{color}$ ) for each category. The output of this process is an extracted car body region ( $CBR_{cat}$ ) depending on the current category ( $cat$ ). The procedure consists of five steps described as follows.

In step 1, if the category is colored, the specular-free operator (**SF**) is executed on the input image ( $Img$ ) and a specular-free image is produced. This operator is not performed in the bright or dark categories. In step 2, the output image of step 1 is segmented by the function **SEG**, as described in Section 3.1, and a segmented image ( $SImg$ ) is produced. Because the bin number of the histogram ( $BinNum$ ) affects the capability of separating the car body and the background, this parameter should be set carefully.

In step 3, the function **partitionSeparate** is executed to generate a new segmentation image ( $SepImg$ ) which clusters the pixels by not only considering the color but also the connectivity relationship. The disconnected partitions with the consistent color region in the original segmented image ( $SImg$ ) are separated.

In step 4, the function **microMerge** is invoked to prevent the generation of numerous mini regions in the new segmented image ( $SepImg$ ). The mini regions in  $SepImg$  are merged with their neighboring larger regions and a new segmented image ( $MergSepImg$ ) is generated. In this step, a very large value is set to the intensity threshold ( $microTH_{int}^{dark}$ ) for dark cars to resist the substantial influences of strong light reflection. For bright and colored cars, the medium values are assigned to the saturation threshold and the intensity threshold ( $microTH_{sat}^{bright}$ ,  $microTH_{int}^{bright}$ ) or ( $microTH_{sat}^{colored}$ ,  $microTH_{int}^{colored}$ ) so that the mini region will not be merged to the large region whose color is obviously different to the color of the mini region, even they are neighboring.

In step 5, the function **macroMerge** is called to construct the whole car body  $CBR_{cat}$ . A seed region is selected first, and one or more partitions in the segmented image ( $MergSepImg$ ) are selected and merged to the seed region if they are connected and have similar colors. The thresholds for color difference on each category are settled differently. For bright or dark cars, a large intensity threshold (empirically,  $macroTH_{int}^{bright} = macroTH_{int}^{dark} = 90$ , if hue, saturation, and intensity values are ranged from 0 to 255) and a small saturation threshold (empirically,  $macroTH_{sat}^{bright} = macroTH_{sat}^{dark} = 30$ ) are settled to collect most car body partitions and not to include the background regions with obvious colors. A considerably large value is assigned to the hue threshold (empirically,  $macroTH_{hue}^{bright} = macroTH_{hue}^{dark} = 255$ ), because the hue value of grayscale color is meaningless. For colored cars, the saturation and hue values are more meaningful than the intensity value. Hence, the small values are defined for  $macroTH_{hue}^{colored}$  and  $macroTH_{sat}^{colored}$ , and a considerably large

value is set to  $macroTH^{colored}_{int}$  (e.g. ( $macroTH^{colored}_{hue}$ ,  $macroTH^{colored}_{sat}$ ,  $macroTH^{colored}_{int}$ ) = (20, 30, 255)). In this step, the number of the merging times ( $MergNum$ ) must also be properly controlled for dark, bright, and colored cars individually to recover the entire car body.

### Procedure: Tri-state Car Body Candidate Generation Process

**Input:**  $Img$ : an input vehicle image

$cat$ : category among bright/dark/colored

$BinNum_{\{bright, dark, colored\}}$ : the bin numbers of color histogram for each category

$MergNum_{\{bright, dark, colored\}}$ : the number of merging times for each category

$microTH^{\{bright, dark, colored\}}_{\{hue, sat, int\}}$ : hue, saturation and intensity thresholds for **microMerge**

$macroTH^{\{bright, dark, colored\}}_{\{hue, sat, int\}}$ : hue, saturation and intensity thresholds for **macroMerge**

**Output:**  $CBR_{cat}$ : the extracted car body region depending on the input category  $cat$

#### Step

```

1 If  $cat == 'colored'$             $I = SF(Img)$ ;
   Else                           $I = Img$ ;                               End
2 If  $cat == 'colored'$             $BinNum = BinNum_{colored}$ ;
   Elseif  $cat == 'bright'$         $BinNum = BinNum_{bright}$ ;
   Else                            $BinNum = BinNum_{dark}$ ;                               End
    $SImg = SEG(I, BinNum)$ ;
3  $SepImg = partitionSeparate(SImg)$ ;
4 If  $cat == 'colored'$             $th_{color} = [microTH^{colored}_{hue}, microTH^{colored}_{sat}, microTH^{colored}_{int}]$ ;
   Elseif  $cat == 'bright'$         $th_{color} = [microTH^{bright}_{hue}, microTH^{bright}_{sat}, microTH^{bright}_{int}]$ ;
   Else                            $th_{color} = [microTH^{dark}_{hue}, microTH^{dark}_{sat}, microTH^{dark}_{int}]$ ;   End
    $MergSepImg = microMerge(SepImg, th_{color})$ 
5 If  $cat == 'colored'$             $th_{color} = [macroTH^{colored}_{hue}, macroTH^{colored}_{sat}, macroTH^{colored}_{int}]$ ;
    $MergNum = MergNum_{colored}$ ;
   Elseif  $cat == 'bright'$         $th_{color} = [macroTH^{bright}_{hue}, macroTH^{bright}_{sat}, macroTH^{bright}_{int}]$ ;
    $MergNum = MergNum_{bright}$ ;
   Else                            $th_{color} = [macroTH^{dark}_{hue}, macroTH^{dark}_{sat}, macroTH^{dark}_{int}]$ ;
    $MergNum = MergNum_{dark}$ ;                               End
    $CBR_{cat} = macroMerge(MergSepImg, th_{color}, MergNum, cat)$ ;

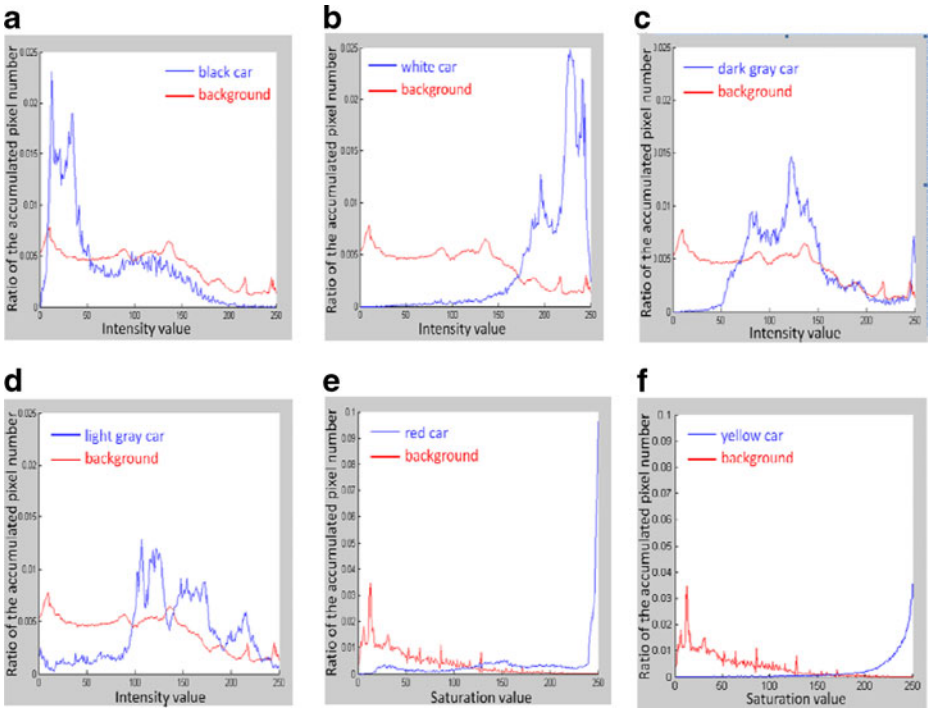
```

List 4 Tri-state car body candidate generation process

### 3.5 Process justification and parameter selection

In this section, process justification and parameter selection based on the characteristics of the color histograms and the spatial relationship of the background and cars of various colors are performed. We justify two critical processes: (1) the strategy of selecting the seed region and (2) the tri-state architecture. We also determine two key parameters: (1) the number of the color bins ( $BinNum$ ) and (2) the number of macro merging times ( $MergNum$ ), according to the requirements of optimization.

In Fig. 5(a)–(f), the color histogram distributions of six cars (blue lines) with black, white, dark-gray, light-gray, red, and yellow colors are demonstrated. Because the grayscale cars can only be distinguished by the intensity, the intensity histograms for the grayscale cars are shown in Fig. 5(a)–(d). Because the saturation reveals useful information to distinguish the colored cars, the grayscale cars, and the background, the saturation histograms for the



**Fig. 5** a–f The intensity a–d and saturation e–f histograms of six cars (blue lines) with black, white, dark-gray, light-gray, red and yellow colors and the background (red lines)

colored cars are shown in Fig. 5(e)–(f). In these figures, the histogram of the background (red line) is averaged over several backgrounds of complex images. Moreover, the values of the blue and red lines in y-axes represent the ratio of the accumulated pixel number that belongs to each intensity (or saturation) value over the total pixel number of the car body or the background.

In Fig. 5(a), the intensity histogram of a black car has a peak region in the low intensity interval. Numerous pixels which belong to the car body and have considerably dark colors contribute to this peak. Therefore, a part of the car body can be included in the top three candidates and selected as the seed region by our proposed seed selection strategy. However, because of the effect of light reflection, the intensity distribution of the car body is widely spread and considerably overlapped with the intensity distribution of the background. If a small number of color bins (*BinNum*) are assigned, some parts of the car body may be easily merged by the background. Therefore, a larger number of color bins (*BinNum*) are required. Under this condition, the car body will be divided into many small pieces. Hence, a larger value must be assigned to the number of macro merging times (*MergNum*) to collect these small pieces and form up the whole car body.

In Fig. 5(b), the intensity histogram of a white car has a peak region in the high intensity interval. Abundant pixels that belong to the car body with very bright color appear in the car image. Therefore, a part of the car body can be included in the top three candidates and selected as the seed region by our proposed seed selection strategy. The intensity distribution of the white car is not spread as widely as that of the black car indicating that the effect of light reflection on white cars does not split the car body into numerous small pieces.

Therefore, a finite number (for example, 3–5) of the macro merging times (*MergNum*) is sufficient to form up the whole car body. However, the intensity distributions of the car body are still overlapped by the distribution of the background. A larger number of color bins (*BinNum*) are required to separate the car body and the background.

The intensity histogram of a dark-gray car and a light-gray car are displayed in Fig. 5(c) and (d). One or more peak regions are observed in these histograms. Under our seed region selection strategy, the region with either the minimal (in dark car category) or the maximal (in bright car category) intensity from the top-three largest regions will be selected as the seed region. When the region with minimal intensity is selected, some large parts which usually appear in the darker interval but belong to the background or undesirable foreground (for example, the shadow, windshield, or bumper) may be selected as a wrong seed. While, if the region with maximal intensity is selected as the white car cases, a correct seed region which is selected from the car body can still be obtained.

In general, the histogram of the background is less distributed in high intensity interval than in low intensity interval, because the effect of light reflection is not as obvious in the background as in the car body. Since only the top-three largest regions in the inner part are considered as the seed candidates, the small regions or the regions which extend to the outer part will not be selected as the seed region, even they have colors with high intensity. Moreover, light reflection usually changes the colors of some car body parts to colors with higher intensity. One of these parts which has the high intensity would belong to the top-three largest regions. Therefore, this region will be correctly selected as the seed region by the proposed seed selection strategy for bright car category.

If we create the fourth state which selects the seed region whose average intensity is mostly close to the middle of the histogram, some parts of the background such as asphalt road or cement wall may easily be selected as a wrong seed region. Furthermore, a bad car body will be determined. Increasing the number of the states does not improve (usually degrade) the performance of the system. Therefore, triple states (i.e. two states for grayscale cars and one state for colored cars) for seed selection are adequate to extract the car body and further improve the color classification.

In some extreme cases where the intensity of a gray car is very dark, the seed selection strategy for bright car may not be suitable. No large parts of the car body with high intensity belong to the top-three largest regions. A wrong seed belonging to the background with middle intensity may be selected. However, the seed selection for dark car category is invoked simultaneously and it may select a more proper seed region. If the very dark and undesirable regions are not included in the top-three largest regions, a part of the car body can be correctly selected as the seed region. Even in the worse case, a considerably dark and undesirable region, such as the shadow, windshield, or bumper, is selected as the seed region. Because these regions are neighboring to the car body and they have colors with similar intensity, the car body can be merged by the seed region and the whole car body can still be recovered; though some undesirable regions are also included. Therefore, triple states strategy for seed selection is sufficient to handle these extreme cases.

From Fig. 5(c) and (d), because the intensity distribution of the car body in dark-gray and light-gray cases is mostly overlapped with those of the backgrounds, a larger number of color bins (*BinNum*) are required. Moreover, a finite number of the macro merging times (*MergNum*) is sufficient to form up the whole car body because the essential parts of the intensity distribution are concentrated.

In the colored car cases as shown in Fig. 5(e) and (f), the saturation distributions of the red and the yellow cars are concentrated and distinguishable from the background after the

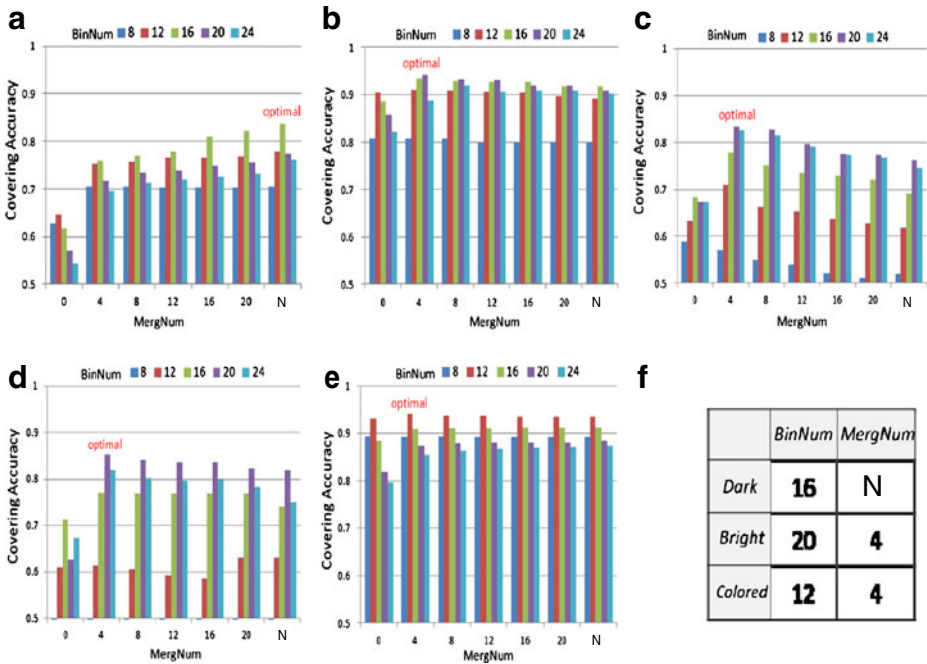


car image is processed through the specular-free processing. The proposed seed selection strategy for colored cars can extract the correct seed by comparing the saturation and selecting the partition with the largest area. To consider the efficiency of the system, a small number of color bins (*BinNum*) and a small number of macro merging times (*MergNum*) are sufficient for the red and yellow cars. In fact, all types of colored cars have concentrated saturation distributions. Therefore, the same processes and parameters are adopted. Single colored category is sufficient to cover all colored car cases.

Next, we analyze the performance in a quantitative manner. The covering accuracy (*CovAcc*), which is defined by Eq. 3, is considered as the performance index. The ground truth of the car body ( $CBR_{gt}$ ) is manually extracted on each vehicle image. The covering accuracy computes the intersection of the extracted car body ( $CBR_{cat}$ ) and the ground truth car body. The parameters which extract the car body with maximal covering accuracy are considered the optimal parameters.

$$CovAcc(CBR_{cat}) = \frac{\# \text{ of } px \in CBR_{cat} \cap CBR_{gt} + \# \text{ of } px \notin CBR_{cat} \cap CBR_{not-gt}}{\# \text{ of } px \in \text{Img}} \quad (3)$$

For the parameter *BinNum*, 8, 12, 16, 20, and 24 are tested, and for the parameter *MergNum*, 0, 4, 8, 12, 16, 20, and N (the number of segmented regions) are tested. The tri-state algorithm is executed by using these parameters on the car images that are stored in our collected datasets. The average covering accuracies on black, white, dark-gray, light-gray, and colored cars are shown in Fig. 6(a)–(e), respectively. The results with optimal parameters are marked in these figures. From Fig. 6(b)–(d), we can see that the optimal parameters in the dark-gray cars and the light-gray cars are identical to those in the white



**Fig. 6** The *CovAcc* analysis with various *BinNum* and *MergNum* parameters for **a** black, **b** white, **c** dark-gray, **d** light-gray, and **e** colored cars. **f** Optimal parameters collection table

cars. As previous analysis, the seed region selection strategies for these three cases are also identical. To simplify the system architecture, we integrated the white, light-gray, and dark-gray cars into one group and termed as the bright car category. The dark car category covers the black cars and the extreme cases of dark-gray cars with very low intensity. All types of colored cars are integrated as the colored car category. Therefore, a tri-state (dark, bright, colored) architecture is proposed. Three states are separately executed with their (nearly) optimal parameters, listed in Fig. 6(f).

#### 4 Car body determination and color classification

After three car body candidates:  $CBR_{bright}$ ,  $CBR_{dark}$ , and  $CBR_{colored}$  have been generated; one of these candidates is selected as the car body. The candidates are tested using two criteria defined in Section 4.1. A hierarchical SVM framework introduced in Section 4.2 classifies the vehicle colors.

##### 4.1 Car body determination

A car body should be the most significant object in a vehicle image. If a region contains large number of pixels included in the inner part, the region is more significant. Hence, Eq. 4 shows a significance factor that is the number of pixels in the inner part minus the number of pixels in the outer part, and the result is divided by the area size of the inner part. If the candidate is not the car body, it may be split into many small pieces. Therefore, a compactness factor, defined in Eq. 5, is utilized to test the scatter degree of each CBR candidate. The scatter degree is related to the number of involved edge pixels of the CBR. The pixel is considered an edge pixel when it at least has a neighboring ( $7 \times 7$ ) pixel not belonging to the CBR. The compactness factor is the ratio of non-edge pixels to all pixels in the CBR candidate. After normalizing the two factors within the range of  $[0, 1]$ , they are multiplied together, and the candidate with the highest score is selected to be the car body as Eq. 6.

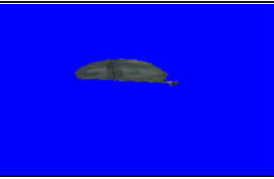
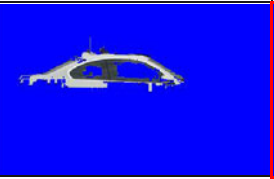

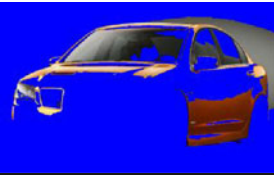

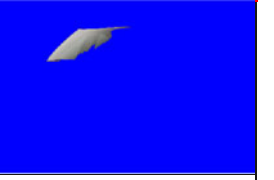


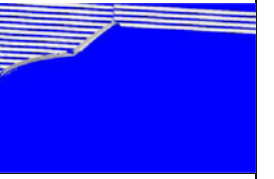
$$Sign(CBR_i) = \frac{\# px \in CBR_i \cap InnerImg - \# px \in CBR_i \cap OuterImg}{\# px \in CBR_i \cap InnerImg}, \quad i \in \{bright, dark, colored\} \quad (4)$$

$$Comp(CBR_i) = \frac{\# px \notin edge(CBR_i)}{\# px \in CBR_i}, \quad i \in \{bright, dark, colored\} \quad (5)$$

$$CBR_{select} = \arg \max_{CBR_i} (Sign(CBR_i) \times Comp(CBR_i)) \quad i \in \{bright, dark, colored\} \quad (6)$$

Table 2 shows the three CBR candidates, and their respective significance and compactness factors for the three cars shown in Fig. 2(a)–(c). For the white car, an improper dark seed from the windshield generates  $CBR_{dark}$ ; therefore, it has low significance. Because the specular-free operator is unsuitable for white cars, the compactness and significance of  $CBR_{colored}$  are small.  $CBR_{bright}$  has good significance and compactness and is selected as the extracted car body. For the orange car, the dark seed in  $CBR_{dark}$  leads to low significance and

**Table 2** The generated CBRs for the white, orange, and black cars in Fig. 2(a)–(c). The significance and compactness factors for each generated CBR are shown below the CBR. For each car, the CBR with the red boundary is selected as the car body region

Category	CBR <sub>dark</sub>	CBR <sub>colored</sub>	CBR <sub>bright</sub>
White car			
	Sign(CBR <sub>dark</sub> ) = 0.3396 Comp(CBR <sub>dark</sub> ) = 0.7227 Sign × Comp = 0.2452	Sign(CBR <sub>colored</sub> ) = 0.4325 Comp(CBR <sub>colored</sub> ) = 0.5794 Sign × Comp = 0.2506	Sign(CBR <sub>bright</sub> ) = 0.7042 Comp(CBR <sub>bright</sub> ) = 0.8052 Sign × Comp = 0.5670
Orange car			
	Sign(CBR <sub>dark</sub> ) = 0.5928 Comp(CBR <sub>dark</sub> ) = 0.6754 Sign × Comp = 0.4004	Sign(CBR <sub>colored</sub> ) = 0.9151 Comp(CBR <sub>colored</sub> ) = 0.8061 Sign × Comp = 0.7377	Sign(CBR <sub>bright</sub> ) = 0.2993 Comp(CBR <sub>bright</sub> ) = 0.7162 Sign × Comp = 0.2144
Black car			
	Sign(CBR <sub>dark</sub> ) = 0.7914 Comp(CBR <sub>dark</sub> ) = 0.8763 Sign × Comp = 0.6935	Sign(CBR <sub>colored</sub> ) = 0.7893 Comp(CBR <sub>colored</sub> ) = 0.6392 Sign × Comp = 0.5045	Sign(CBR <sub>bright</sub> ) = 0.0899 Comp(CBR <sub>bright</sub> ) = 0.2946 Sign × Comp = 0.0265

low compactness. Since the specular-free operator is adopted, the significance and compactness of  $CBR_{colored}$  is evidently higher to those in  $CBR_{bright}$ , enabling selection of  $CBR_{colored}$ . For the black car, the specular-free operator destroys the image, so the compactness of  $CBR_{colored}$  is small. In addition, an improper seed generates  $CBR_{bright}$ , so that the significance and the compactness are poor. By having a proper seed and merging process,  $CBR_{dark}$  displays good significance and compactness and is selected as the extracted car body for the black car.

#### 4.2 Vehicle color classification

Numerous vehicle color classification methods have been proposed in the past decade. HSV color model reduces the effect of circumstance luminance compared to RGB color space [22], and SVM shows excellent capability for identifying features [5, 24]. Although some studies claim a good successful rate, their performances are still influenced by impure background pixels and improper foreground pixels [25]. Since this study executed a beneficial car body extraction algorithm as described in the previous sections, the color classification methods can achieve high performance by only considering the selected car body.

For each pixel in CBR, the RGB color value ( $R_{CBR}, G_{CBR}, B_{CBR}$ ) is converted to the HSV color value ( $H_{CBR}, S_{CBR}, V_{CBR}$ ) before computing the 3D HSV color histogram of the pixels in the CBR. Figure 7 shows the hierarchical SVM classifier to identify the color type of the extracted car body. A binary SVM classifier:  $SVM^2_{CG}(S_{CBR})$  is performed on the  $S_{CBR}$  (Saturation) channel to distinguish the colored cars from the grayscale cars in the first layer. In the second layer, the more precise color types are going to be classified. If the category is colored, a multi-class SVM:  $SVM^M_{colored}(H_{CBR})$  is performed on the  $H_{CBR}$  (Hue) channel to classify the car body into one of the colored types: red, orange, yellow, green, blue, purple, and pink. If the category is grayscale, the other multi-class SVM:  $SVM^M_{WSB}(V_{CBR})$  is performed on the  $V_{CBR}$  (Value) channel to classify the car body into one of the grayscale types: white, gray, and black.

### 5 Experimental results

#### 5.1 Experimental description

Two other car body extraction and color classification methods were implemented for comparison to reveal the advantages of the proposed algorithm. Three algorithms are tested and compared in the experiments.

- 1) *Homogeneous SVM (homo-SVM) method*: After the image is segmented into multiple regions, the largest part in the inner part, in which all its pixels have similar colors, is selected. This part is subsequently fed into the SVM [2] to classify its color. Through the process for parameter selection, as in the proposed method described in Section 3.5, the number of color bins (*BinNum*) used in this method is selected as 12.
- 2) *Removal rule method* [25]: After the image is segmented into multiple regions, the dominant color of the central part is computed. The segmented regions are subsequently excluded with average colors that differ from the dominant color, and the regions with an excessively large or small average intensity are removed. Finally, the remaining regions are grouped as the car body and the color is classified based only on the pixels within the car body. Through the process for parameter selection, the number of color bins (*BinNum*) used in this method is selected as 16.
- 3) *Tri-state method*: The proposed method of this study. The procedure of the tri-state framework is performed as shown in Fig. 1. This method provides distinct strategies for

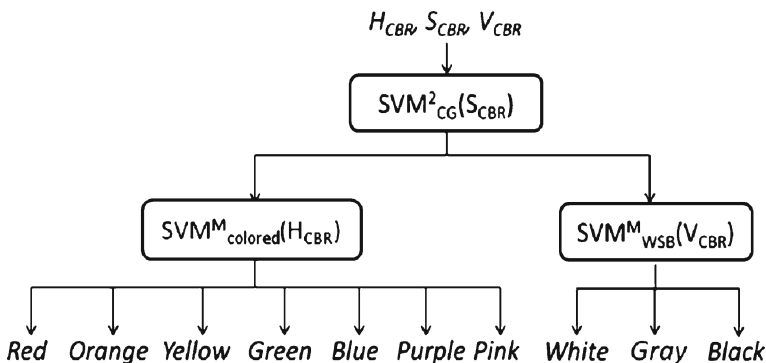


Fig. 7 Hierarchical SVM of color type classification

selecting seeds from bright, dark, and colored cars. In addition, the seed region is not confined to similar color to the dominant color of the central part of the image. Anti-reflection design (SARM algorithm) is also provided to enhance the performance of car body extraction and color classification. Through the process of parameter selection, the number of color bins (*BinNum*) and the number of macro merging times (*MergNum*) are selected according to the results decided in Section 3.5.

The above three methods were implemented on an AMD Athlon II X2 240, 2.0 GB RAM, 2.81 GHz PC with a Language C environment and working on the datasets as described in Section 5.2. The four criteria to evaluate the performances of car body extraction and color classification are provided in Section 5.3. Experimental data was collected according to the requirement of the evaluation criteria. Finally, in Section 5.4, the performances of these three methods are evaluated and compared based on the experimental results.

## 5.2 Collected datasets

To test the proposed tri-state method and the two compared methods on the intelligent transportation system (ITS) and the content-based image retrieval (CBIR) system, this study collected two vehicle image datasets for the ITS and CBIR systems. In addition, the CBIR dataset is regrouped into two subsets depending on the complexity of the background

The first dataset captures car images from real traffic streams obtained using cameras, to simulate the ITS scenario. The camera is placed at a fixed position; hence the view of the captured vehicle is usually fixed or known. Because of the static background, a background subtraction technique [6] is adopted. The vehicle images in this dataset include only the interesting foreground regions of the vehicles. The viewpoint constraint and the background subtraction facilitate achieving the vehicle classification, and most studies test their methods in this scenario. Seven common colors are present in the dataset: white, black, gray, red, yellow, green, and blue. Each color set contains 100 car images. Table 3 displays several sample images from this dataset.

To simulate the CBIR scenario, the second dataset collects car images from the Internet. Since vehicle images on the Internet may be captured from any viewpoint and the background model cannot be constructed, vehicle color classification in this scenario is more challenging. In addition, not only the seven color type cars exist as in ITS dataset, but the cars with three rare color types: pink, purple, and orange, are also collected in this dataset. Each color type contains about 30 car images, and the total number of images is over 310. We further divide these images into two equivalent quantity subsets. One subset contains the images with more complicated backgrounds, such as complicated building. Furthermore, car bodies may not be certainly located at the central part, or not guarantee to occupy a sufficient large partition of the image. We term this dataset as the CBIR-complex dataset. The rest images are collected as the CBIR-simple dataset. Several sample images of these two datasets are listed in Table 4.

In the experiments, the ten-fold cross validation is adopted to evaluate the performances. In other words, each dataset of ITS, CBIR-simple, and CBIR-complex is divided into ten subsamples. When each dataset is tested, the operation of using one subsample for testing and the other nine subsamples for training is repeated for ten times. The successful rates for color classification are averaged over all the subsamples. The datasets and the implementation of the proposed system are available at [9].

**Table 3** Sample images from the ITS dataset

Color	ITS				
Black					
Gray					
White					
Red					
Yellow					
Green					
Blue					

### 5.3 Performance criteria

Four criteria: covering ratio ( $CovRat$ ), color deviation ( $ColDev$ ), classification accuracies of color category ( $AC_{cat}$ ), and color type ( $AC_{ColorType}$ ) are defined to evaluate the performances of the aforementioned three algorithms.

Covering ratio which is defined as the percentage of pixels covering both the extracted car body  $CBR_{cat}$  and the ground truth car body  $CBR_{gt}$  can be taken as the performance index. The covering ratio comprises the covering precision ( $CovPrec$ ) and the covering recall ( $CovRec$ ), as defined by Eqs. 7 and 8. The covering precision shows the percentage of the pixels in the extracted car body belongs to the real car body. And, the covering recall shows the percentage of the pixels in the real car body has been extracted.

$$CovPrec(CBR_{cat}) = \frac{\# \text{ of } px \in CBR_{cat} \cap CBR_{gt}}{\# \text{ of } px \in CBR_{cat}} \quad (7)$$

$$CovRec(CBR_{cat}) = \frac{\# \text{ of } px \in CBR_{cat} \cap CBR_{gt}}{\# \text{ of } px \in CBR_{gt}} \quad (8)$$

**Table 4** Sample images from the CBIR-simple and CBIR-complex datasets

	CBIR-simple	CBIR-complex
Black		
Gray		
White		
Red		
Orange		
Yellow		
Green		
Purple		
Blue		
Pink		

Covering accuracy (*CovAcc*) used in Section 3.5 is also a proper criterion to evaluate the covering ratio. Since the covering precision and the covering recall reveals different information on car body extraction, they are both adopted for performance evaluation.

Color deviation (*ColDev*) shows the average color difference between the extracted car body  $CBR_{cat}$  and the ground truth car body  $CBR_{gt}$ . Manually select  $k$  pixels spreading in the  $CBR_{cat}$  to attain the hue ( $H^1_{gt}, \dots, H^k_{gt}$ ) and intensity ( $I^1_{gt}, \dots, I^k_{gt}$ ) values of these pixels as the ground truth colors of the car. For colored cars, the color deviation is computed by the average difference between the hue values of the pixels ( $px$ ) in the  $CBR_{cat}$  and the ground truth hue, as shown in Eq. 10. For bright or dark cars, color deviation (*ColDev*) is computed by the average difference between the intensity values of the pixels in the  $CBR_{cat}$  and those of the ground truth intensity, as shown in Eq. 11.

$$ColDev(CBR_{cat}) = \begin{cases} HueDev(CBR_{cat}) & , \text{ if } cat \in \text{colored} \\ IntDev(CBR_{cat}) & , \text{ if } cat \in \text{bright or dark} \end{cases} \quad (9)$$

$$HueDev(CBR_{cat}) = \frac{\sum_{px \in CBR_{cat}} \min_{1 \leq i \leq k} (|Hue(px) - H_{gt}^i|)}{\# px \in CBR_{cat}} \tag{10}$$

$$IntDev(CBR_{cat}) = \frac{\sum_{px \in CBR_{cat}} \min_{1 \leq i \leq k} (|Int(px) - I_{gt}^i|)}{\# px \in CBR_{cat}} \tag{11}$$

The accuracies of color category ( $AC_{cat}$ ) and color type ( $AC_{ColorType}$ ) demonstrate the performance of color classification. For each image in a dataset, the color category and the color type are decided using several support vector machines (SVM). Two binary SVM:  $SVM^2_{CG}$  and  $SVM^2_{BD}$  compose a function  $SVM_{cat}$  to determine the color category of the extracted  $CBR_{cat}$ , as shown in Eq. 12. The SVM:  $SVM^2_{CG}$  distinguish colored cars from grayscale cars by saturation histogram. The SVM:  $SVM^2_{BD}$  separates bright and dark cars by intensity histogram. A color type decision function  $SVM_{ColorType}$  comprises two multi-class SVM:  $SVM^M_{colored}$  and  $SVM^M_{WSB}$ , to determine the color type of the extracted CBR, as shown in Eq. 13. The SVM:  $SVM^M_{colored}$  identifies multiple colors from the colored category. The SVM:  $SVM^M_{WSB}$ , further recognizes white, gray and black cars from the grayscale vehicles.

$$AC_{cat} = \frac{\sum_{i=1}^{\# \text{ of images}} SVM_{cat}(CBR_i) == SVM_{cat}(CBR_i^{gt})}{\# \text{ of images in the dataset}} \tag{12}$$

$$AC_{ColorType} = \frac{\sum_{i=1}^{\# \text{ of images}} SVM_{ColorType}(CBR_i) == SVM_{ColorType}(CBR_i^{gt})}{\# \text{ of images in the dataset}} \tag{13}$$

### 5.4 Performance evaluation

This section presents the evaluation of the performance metric: (1) accuracy of color category classification, (2) covering ratio for car body, (3) color deviation, (4) accuracy of color type classification and (5) computation load.

#### (1) Accuracy of Color Category Classification

Table 5 shows the confusion matrices of color category classification of the proposed tri-state method performing on the ITS, CBIR-simple and CBIR-complex datasets. In the ITS dataset, the car images with categories of 230 bright, 170 dark and 400 colored are

**Table 5** Confusion matrices of color category classification on ITS, CBIR-simple and CBIR-complex datasets

T/D	ITS				CBIR-simple				CBIR-complex			
	Bright	Dark	Colored	AC <sub>cat</sub>	Bright	Dark	Colored	AC <sub>cat</sub>	Bright	Dark	Colored	AC <sub>cat</sub>
Bright	228	0	2	99.1%	30	0	0	100.0%	30	0	0	100.0%
Dark	0	170	0	100.0%	0	20	0	100.0%	1	19	0	95.0%
Colored	0	0	300	100.0%	1	0	104	99.1%	3	0	102	97.1%



tested. Only two bright cars are misclassified into the colored category, and all dark and colored cars are correctly classified into their categories. The average accuracy of color category classification on the ITS dataset is 99.7%. In the CBIR-simple dataset, 30 bright, 20 dark, and 105 colored car images are classified. Only 1 colored car is misclassified by the tri-state method. The average accuracy of color category classification on the CBIR-simple dataset is 99.3%. In the CBIR-complex dataset, 30 bright, 20 dark and 105 colored car images are classified their categories. Only 1 dark car and 3 colored cars are misclassified. The average accuracy of color category classification on the CBIR-complex dataset is 97.4%.

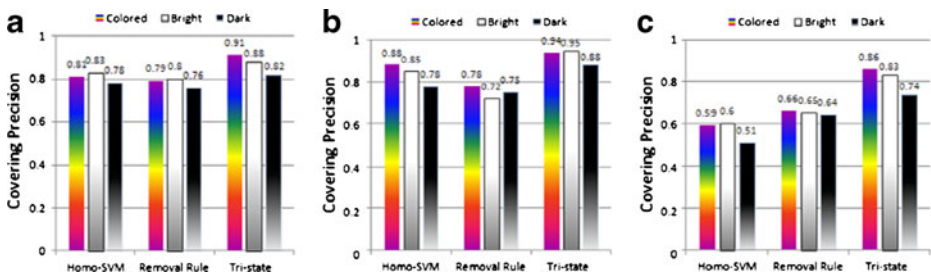
(2) Covering Ratio for Car Body

Figures 8 and 9 show the covering precisions and the covering recalls of the homo-SVM method, the removal rule method, and the tri-state method on the ITS, CBIR-simple, and CBIR-complex datasets, respectively.

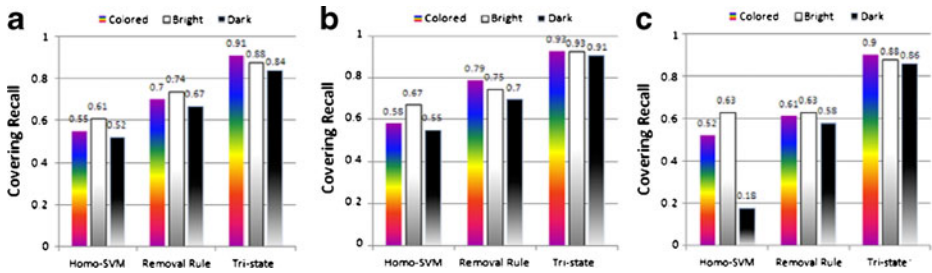
In Fig. 8(a), the covering precisions of the three methods are tested on the ITS dataset. Because most of the background pixels are subtracted in this scenario, the car body can be detected more easily so that all the three methods can provide satisfactory covering precisions. The average covering precisions of the homo-SVM, removal rule method, and tri-state approaches are 81%, 78% and, 87%, respectively. The tri-state method is slightly superior to the other two methods. In Fig. 9(a), the average covering recalls of these three methods are 56%, 71% and, 88%, respectively. The average covering recall of the homo-SVM is inferior to that of the other approaches because only the most significant part is selected. The tri-state method has smart merging process with (nearly) optimal parameters for various color car category, the average covering recall of the tri-state method is obviously superior to the other two approaches.

In Figs. 8(b) and 9(b), the three compared methods are tested on the CBIR-simple dataset. The average covering precisions of these three methods are 81%, 75% and, 92%, respectively. And, the average covering recalls of these three methods are 60%, 75% and, 92%, respectively. The performances of three methods in this dataset are similar to those in the ITS dataset. The tri-state method provides significantly superior covering recall to the other two methods.

Figure 8(c) shows the covering precisions of the three compared methods on the CBIR-complex dataset. In this scenario, the multi-colored background is included in the image; therefore, the regions with the same color to the vehicle are easily falsely distinguished as the car body by the homo-SVM method. Hence, the average covering precisions of the homo-SVM method decrease to 57%. In addition, the car body in this scenario may not completely appear in the central part of the image. Therefore, the



**Fig. 8** Covering precisions of three compared methods on **a** ITS, **b** CBIR-simple, and **c** CBIR-complex datasets for each color category



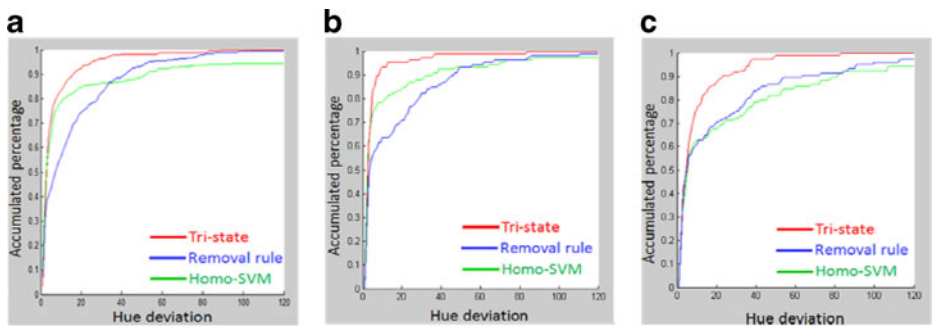
**Fig. 9** Covering recalls of three compared methods on **a** ITS, **b** CBIR-simple, and **c** CBIR-complex datasets for each color category

average covering precisions of the removal rule method only achieve 65%. The average covering precisions of the tri-state method are 81%. Although the covering precision of the tri-state method in the CBIR-complex scenario is slightly lower than those in the ITS and CBIR-simple scenarios, its superiority to the other methods is very obvious. In Fig. 9(c), we see that the covering recalls of the homo-SVM method and the removal rule method are only 45% and 61%, respectively. However, the average covering recall of the tri-state method can still achieve 88%, even in this CRIR-complex scenario. We conclude that the tri-state method can provide significantly superior covering precision and covering recall to the homo-SVM method and the removal rule method on the ITS, CBIR-simple, and CBIR-complex scenarios.

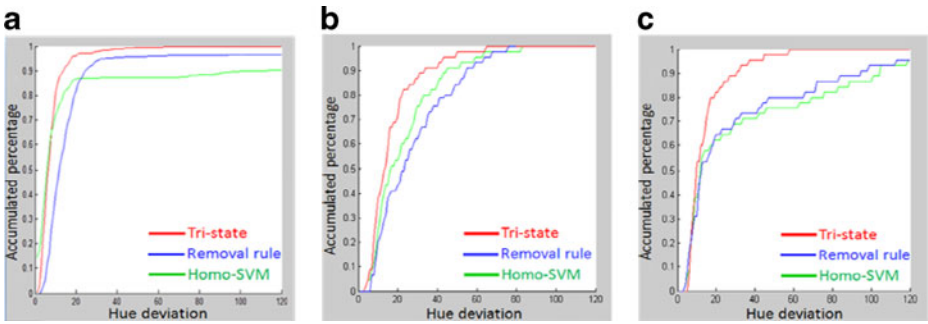
(3) Color Deviation

The extracted car body may involve numerous pixels that do not belong to the actual car body. Hence, the color histogram may be deviated by these pixels. The color deviation, as defined in Eqs. 9–11, represent the purity of color in the extracted car body. The hue deviations of the car bodies that are extracted from the three methods are computed for the colored car cases on the ITS, CBIR-simple and CBIR-complex datasets. The intensity deviations are also computed for the grayscale car cases. The accumulated percentage of these car bodies for colored and grayscale cars are shown in Figs. 10 and 11, respectively.

Figure 10(a) shows the hue deviation when the ITS dataset are tested for the three compared methods. If the hue deviation smaller than 30 is tolerable, the accumulated percentages of hue deviation by the homo-SVM method, the removal rule method, and the tri-state method are 85%, 82%, and 96%, respectively. When the tolerance of the



**Fig. 10** Accumulated percentage of hue deviation by three compared methods on colored car images in **a** ITS, **b** CBIR-simple and **c** CBIR-complex datasets



**Fig. 11** Accumulated percentage of intensity deviation by three compared methods on grayscale car images in **a** ITS, **b** CBIR-simple and **c** CBIR-complex datasets

hue deviation increases to 60, the accumulated percentages by these three methods are 91%, 93%, and 98%, respectively. The performance of the homo-SVM method is superior to that of the removal rule method when more colors are required to be classified. If only a few types of colors, such as four colors: red, blue, yellow and green, to be required for classification, the removal rule method is superior to the homo-SVM method. The tri-state method extracts purer colors than the homo-SVM method and the removal rule method so that the tri-state method has better performance on hue deviation.

Figure 10(b) shows the hue deviation when the CBIR-simple dataset are tested for the three compared methods. If the hue deviation smaller than 30 is tolerable, the accumulated percentages of hue deviation by the homo-SVM method, the removal rule method, and the tri-state method are 86%, 82%, and 96%, respectively. When the tolerance of the hue deviation increases to 60, the accumulated percentages by these three methods are 92%, 93%, and 99%, respectively. Because the removal rule method may obtain an incorrect dominant color due to the various viewpoints in CBIR scenario, the accumulated curve increases slower than the curve in the ITS scenario. The homo-SVM extracts purer color car body than the removal rule method because the background is more easily distinguished. The tri-state method selects more proper seeds than the other two methods and it can work well without the limitation of viewpoint. Hence, the improvement of the tri-state method to the other approaches is notable in the CBIR-simple dataset.

Figure 10(c) shows the hue deviation when the images in the CBIR-complex dataset are tested for the three compared methods. If the hue deviations smaller than 30 is tolerable, the accumulated percentages of hue deviation by the homo-SVM method, the removal rule method, and the tri-state method are 74%, 78%, and 90%, respectively. When the tolerance of the hue deviation increases to 60, the accumulated percentages of hue deviation by these three methods are 85%, 89%, and 99%, respectively. Because of the interference of the complicated background, much more incorrect seed region are selected by the removal rule method and the homo-SVM method. The tri-state method generates the triple car body candidates by proper strategies with (nearly) optimal parameters for various color car category. Therefore, the car body can be extracted more accurately and purer colors which approximate the real colors of tested cars are obtained. In this CRIR-complex scenario, the superiority of the tri-state method to the other two compared methods is very obvious.

The grayscale cars are also tested on ITS, CBIR-simple, and CBIR-complex datasets and their intensity deviations are shown in Fig. 11(a)–(c), respectively. In Fig. 11(a), excluding the considerably small intensity deviation cases, the tri-state method is consistently superior to the other approaches. The removal rule method is superior to the homo-SVM method, when the tolerable intensity deviation is larger than 15. While in Fig. 11(b), the removal rule is worse than the other methods due to the effect of viewpoint varying. In Fig. 11(c) the homo-SVM method and the removal rule method are competitive and both of them obtain poor performance. The tri-state method is consistently superior to these two compared methods. By comparing Fig. 10(c) and 11(c), the superiorities of the tri-state method to the two compared methods are even more significantly in the grayscale car cases than in the colored car cases when they work on the CBIR-complex dataset.

#### (4) Accuracy of Color Type Classification

After the car body region has been extracted, the color histogram of the car body is computed and then the SVM function depicted in Fig. 7 is utilized to classify the color type. Figure 12(a)–(c) show the accuracies of the color type classification on the extracted CBR by the three compared methods on ITS, CBIR-simple and CBIR-complex datasets, respectively.

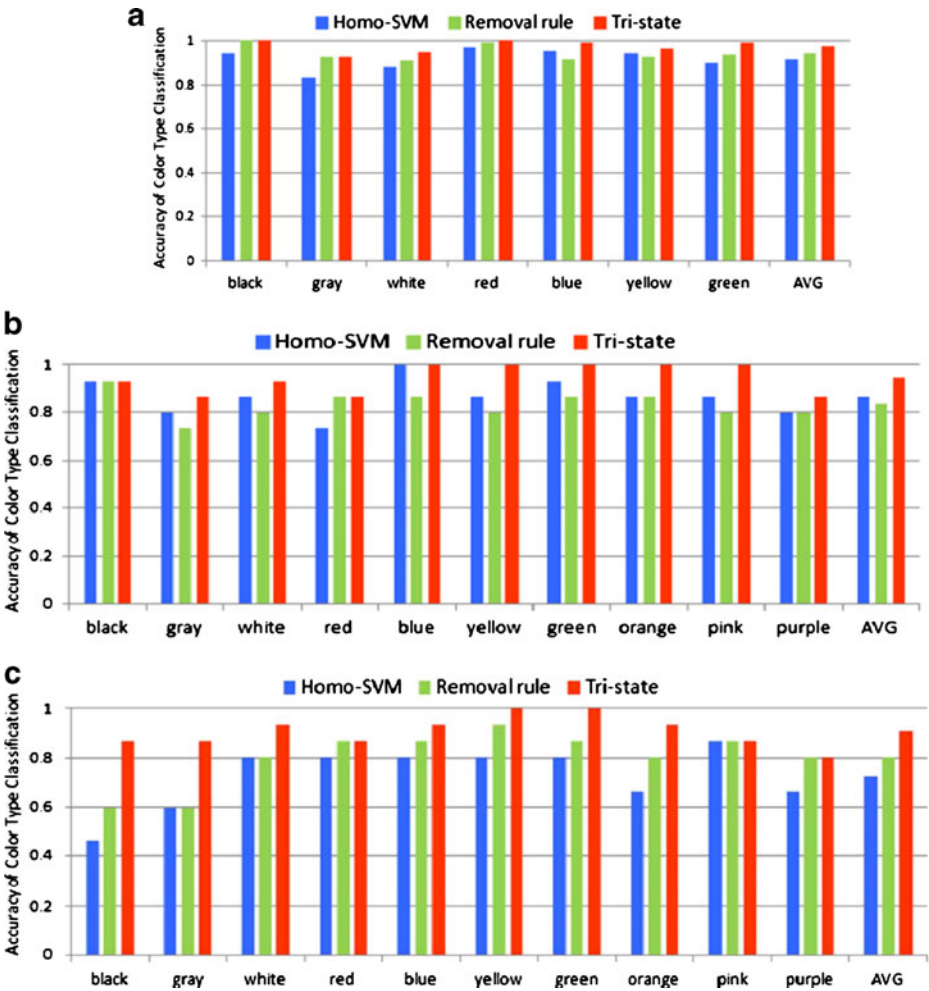
In Fig. 12(a), the average accuracies of color type classification by the homo-SVM method, the removal rule method, and the tri-state, method on the ITS dataset are 91%, 94%, and 97%, respectively. For the colored cars in ITS scenario, because only four kinds of colors: red, blue, green and yellow, are needed to be classified, the hue deviation tolerance can be large. For the grayscale images, only three kinds of colors: black, gray and white, are needed to be classified, the intensity deviation can also be large. All methods perform well under a large tolerance, but the tri-state method still provides approximately 6% and 3% improvements of classification accuracy compared with the homo-SVM method and the removal rule method, respectively.

In Fig. 12(b), there are ten color types to be classified in the CBIR-simple dataset; hence, the tolerances of hue deviation should be smaller. The accuracies of color type classification will get lower and the removal rule method works the worst. From Fig. 9(b), the average accuracy of color type classification by the tri-state method is 94%. The superiorities of the proposed tri-state method to the homo-SVM method and the removal rule method are more significant and they are over 8% and 11%, respectively.

In Fig. 12(c), ten color type cars are classified in the CBIR-complex dataset. Due to the interference of complicated backgrounds, all methods in this dataset perform obviously worse than those in the other datasets. However, the average accuracy of color type classification by the tri-state method can still achieve 91% and the average improvements of the tri-state method to the homo-SVM method and the removal rule method are over 18% and 10%, respectively.

#### (5) Computation Load

The average computation time of each component on an image with  $100 \times 100$  pixels is reported in Table 6. The entire computation time in the proposed system is approximately 0.057 s per image. This length of computation time is applicable to most real-time systems for car body extraction and color classification. The computation time of the two compared algorithms: the homo-SVM and the



**Fig. 12** Accuracies of color type classification of the three compared methods on in **a** ITS, **b** CBIR-simple and **c** CBIR-complex datasets

**Table 6** Average computation time of each component in the proposed system and in the compared methods

Method	Image segmentation	Candidates generation	Car body determination	Color type classification	Whole system
Homo-SVM	12 ms	NULL	1.4 ms	1.6 ms	15 ms
Removal rule	18 ms	NULL	1.4 ms	1.6 ms	21 ms
Tri-state	colored: 12 ms bright: 23 ms dark: 18 ms	colored: 0.28 ms bright: 0.28 ms dark : 0.42 ms	1.8 ms	1.6 ms	57.38 ms

removal rule methods are 0.015 s and 0.021 s, respectively. Although the tri-state method requires slightly more computation time than the two compared methods, it provides a more intact car body and higher color classification accuracy than the two compared methods.

## 6 Conclusion

This study develops an algorithm with a tri-state architecture and including a SARM algorithm to effectively extract the car body and classify the vehicle color in challenging cases with unknown car type, unknown viewpoint, and non-homogeneous light reflection conditions. The characteristics of color histogram and the spatial relationship of the background and cars of various colors are considered. The serious effects of the non-homogeneous light reflection on the car body can be overcome by the proposed algorithm which uses different strategies designed for three critical color categories and the critical parameters for each category are individually selected according to the requirements of optimization. Without viewpoint and car type constrains, all color type cars can be processed optimally with high performance and intact car body extraction and accurate color classification can be obtained simultaneously for wide applications. The computation time of the proposed method is limited. Therefore, it is applicable for real-time systems.

According to the experimental results, our proposed algorithm can obtain satisfactory performance for covering ratio of the car body, deviation of color estimation, and accuracies of color category and color type classifications. The accurate color estimation and color classification would be useful for ITS and CBIR applications. In our current research, the extracted car body also provides useful information regarding numerous image processing issues on vehicles, such as car model recognition and car shape feature extraction.

**Acknowledgements** This work was partially supported by National Science Council grant NSC98-2221-E-009-091-MY3: Multiview multimedia content analysis, indexing and query

## References

1. Al Aghbari Z, Al-Haj R (2006) Hill-manipulation: an effective algorithm for color image segmentation. *Image Vis Comput* 24(8):894–903
2. Baek N, Park S-M, Kim K-J, Park S-B (2007) Vehicle color classification based on the support vector machine method. *Commun Comput Info Sci* 2(24):1133–1139
3. Brown LM (2010) Example-based color vehicle retrieval for surveillance. *International Conference on Advanced Video and Signal Based Surveillance* 91–96
4. Butzke M, Silva AG, Hounsell MS, Pillon MA (2008) Automatic recognition of vehicle attributes – color classification and logo segmentation. *HIFEN* 32(62)
5. Chang C-C, Lin C-J (2001) Training nu-support vector classifiers: theory and algorithms. *Neural Comput* 13(9):2119–2147
6. Chen B, Lei Y, Li W (2004) A novel background model for real-time vehicle detection. *Int Conf Signal Proces* 2:1276–1279
7. Chen Z-Z, Pears N, Freeman M, Austin J (2009) Road vehicle classification using support vector machines. *IEEE International conference on Intelligent Computing and Intelligent Systems*. 214–218
8. Colormap by MathWorks, <http://www.mathworks.com/help/techdoc/ref/colormap.html>
9. Dataset and implementation of the tri-state system: <http://www.cs.nctu.edu.tw/~hcku/>

10. Elgammal A, Harwood D, Davis L (2000) Non-parametric Model for Background Subtraction,” 6th European Conference on Computer Vision, Lecture Notes in Computer Science 1843:751–767
11. Finlayson GD, Schaefer G (2001) Solving for color constancy using a constrained dischromatic reflection model. *Int J Comput Vis* 42(3):127–144
12. Haralick RM, Shapiro LG (1992) Computer and robot vision, Volume I, Addison-Wesley, pp 28–48
13. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):316–323
14. Kim K-J, Park S-M, Choi Y-J (2008) Deciding the Number of Color Histogram Bins for Vehicle Color Recognition. *IEEE Asia-Pacific Services Computing Conference* 134–138
15. Kim S, Park S, Kim M (2003) Central object extraction for object-based image retrieval. *International Conference on Image and Video Retrieval* pp 39–49
16. Li X, Zhang G, Fang J, Wu J, Cui Z (2010) Vehicle color recognition using vector matching of template. *Proc. of International Symposium on Electronic Commerce and Security* 189–193
17. Lin S, Li Y, Kang SB, Tong X, Shum HY (2002) Diffuse-specular Separation and Depth Recovery from Image Sequences. *Proc. of Europe Conference of Computer Vision* 210–224
18. Ohashi T, Aghbari Z, Makinouchi A (2003) Hill-climbing algorithm for efficient color-based image segmentation. *IASTED International Conference on Signal Processing, Pattern Recognition, and Applications* 17–22
19. Santos D, Correia PL (2009) Car recognition based on back lights and rear view features. *Workshop on Image Analysis for Multimedia Interactive Services*. 137–140
20. Shih FY, Cheng S (2005) Automatic seeded region growing for color image segmentation. *Image Vis Comput* 23:877–886
21. Son J-W, Park S-B, Kim K-J (2007) A convolution kernel method for color recognition. *International Conference on Advanced Language Processing and Web Information Technology* 242–247
22. Sural S, Zian G, Pramanik S (2002) Segmentation and histogram generation using the HSV color space for image retrieval. *Int Conf Image Proces* 2:589–592
23. Tan RT, Ikeuchi K (2003) Separating reflection components of textured surfaces using a single image. *IEEE Int Conf Comput Vis* 2:870–877
24. Theodoridis S, Koutroumbas K (2008) *Pattern recognition 4<sup>th</sup> edn*. Academic Press, ISBN: 1597492728, pp 119–128
25. Wu Y-T, Kao J-H, Shih M-Y (2010) A vehicle color classification method for video surveillance system concerning model-based background subtraction. *Advances in Multimedia Information Processing – PCM Lecture Notes in Computer Science* 6297:369–380
26. Xu Z, Cao J (2009) Vehicle color extraction based on first sight window. *IEEE International Conference on Information Science and Engineering* 1503–1506
27. Zhang DS, Lu G (2001) Segmentation of moving objects in image sequence: a review. *Circuits Syst Signal Process* 20(2):143–183



**Hui-Zhen Gu** received her MS degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 2007. She is currently a Ph. D. candidate of Computer Science and Information Engineering in National Chiao Tung University, Taiwan. Her research interests include computer vision, content-based video retrieval, multimedia information system, and intelligent surveillance system.



**Suh-Yin Lee** is a Professor at Computer Science and Information Engineering in National Chiao Tung University, Taiwan. She received her M.S. degree in computer science from University of Washington, Seattle, U.S.A., in 1975, and her Ph.D. degree in computer science from Institute of Electronics, National Chiao Tung University, Taiwan. Her research interests include content-based indexing and retrieval, distributed multimedia information system, mobile computing, and data mining.