A Traffic Sign Detection Method with High Accuracy and Efficiency

Gangyi Wang, Guanghui Ren, Taifan Quan
School of Electronics and Information Technology
Harbin Institute of Technology
Harbin, Hei longjiang, P.R. China
ghui.ren@gmail.com

***Abstract*—Traffic sign recognition is a challenging computer vision problem for its high requirement of reliability and efficiency. In this paper, we propose a prohibitory traffic sign detection method. The method first extracts a red bitmap from the input image and then detects circles in the red bitmap. To detect circles more efficiently, a new circle detection method is proposed which is of high accuracy, low computational complexity and low memory consumption. Experimental results indicate that the proposed method can robustly detect prohibitory signs in all kinds of adverse situations including bad lighting condition, similar background color, multiple signs clustering, and partial occlusion. In addition, the method is efficient enough for real-time applications.**

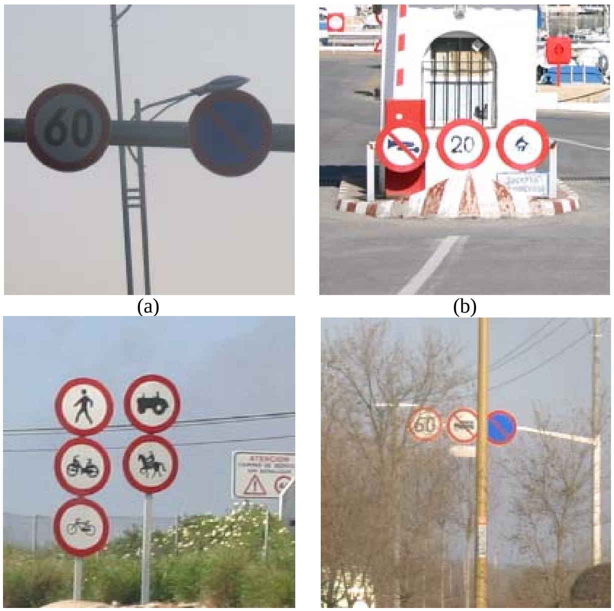
Keywords-traffic sign recognition; circle detection; object detection; real-time

I. INTRODUCTION

Traffic sign recognition is an important function for driver assistant systems. Although many efforts have been made to this field, there is still not a satisfactory traffic sign recognition method. The major difficulties include bad lighting condition, similar background color, multiple signs clustering, and partial occlusion, as shown in Fig. 1.

A traffic sign recognition method usually consists of two steps: detection and classification. The former step finds out all the regions containing traffic signs, and the latter step determines the classes of the signs. Methods for classification have shown great success in recent researches [1], but detection methods are still not satisfactory in complex traffic scenes. Therefore, we focus on traffic sign detection method in this paper.

One important characteristic for traffic signs is the bright and fixed color. Many traffic sign detection methods segment the input images with the color of traffic signs to reduce the searching region. As colors change significantly under different lighting conditions, it is important for color segmentation methods to be invariant to lighting condition. Some researchers [2-4] use the hue-saturation-intensity (HSI) color space, which is based on human color perception, and achieve good robustness to lighting condition. However, the accuracy of these methods suffers from the cases of similar background color and multiple signs clustering, which make the signs hard to be segmented with color.



(a) (b)

(c) (d)

Figure 1. Difficulties for traffic sign recognition. (a) bad lighting condition (b) similar background color (c) multiple signs clustering (d) partial occlusion

Another major characteristic of traffic signs is the regular shape. Hough transform is a commonly used method in detecting regular shapes such as circle and triangle. Some researches use Hough transform [5, 6] to detect circles in an extracted edge bitmap, but the memory consumption of the methods is huge for large images. Some other researchers [7, 8] adopt genetic algorithm to detect circles and achieve high robustness to rotation and scale. However, these methods are too slow for practical use.

In this paper, we propose a new prohibitory traffic sign detection method by utilizing both the color and shape characteristics. As all the prohibitory traffic signs are circular and have red rims, we extract red pixels with a HSV thresholding method, which significantly reduces the searching area. To detect circles in the red edge bitmap, we propose a new circle detection method which is fast and robust to noise and partial occlusion, while the memory consumption is much smaller than Hough transform. We test the method with images from real traffic scenes and prove that the proposed traffic sign detection method is robust to different lighting condition, similar background color, multiple signs clustering, and partial occlusion.

II. THE TRAFFIC SIGN DETECTION METHOD

The proposed method consists of four steps, as shown in Fig. 2. First, all red pixels are extracted from the input image with a HSV thresholding method, and then an edge detector is employed to extract an edge bitmap from the red bitmap. Next, circles in the edge bitmap are detected followed by an extra verification step. Regions pass the verification are finally output as region of interests. Details of each step are described in the following sections.

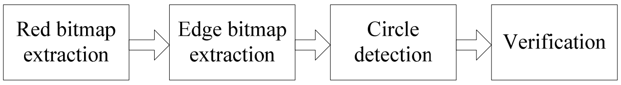


Figure 2. Flow diagram of the proposed method

A. *Red bitmap extraction*

As described in the previous section, the HSI color space is based on human color perception, thus color information can be easily extracted from the hue and saturation components without interfered by lighting condition. However, the input images are represented with red-green-blue (RGB) color space in most cases, and the computational complexity is relatively high to convert the color space from RGB to HSI. So we choose the hue-saturation-value (HSV) color space instead, which is a similar to HSI but with light computational complexity. The equations to convert RGB to HSV are as follows:

$$\mathbf{H}=\begin{cases}(6+\dfrac{\mathbf{G}-\mathbf{B}}{\mathbf{MAX}-\mathbf{MIN}}\ \ \mathrm{mod}\ \ 6)\cdot 60^{\circ},\ \ \text{if } \mathbf{R}=\mathbf{MAX}\\ (2+\dfrac{\mathbf{B}-\mathbf{R}}{\mathbf{MAX}-\mathbf{MIN}})\cdot 60^{\circ},\ \ \text{if } \mathbf{G}=\mathbf{MAX}\\ (4+\dfrac{\mathbf{R}-\mathbf{G}}{\mathbf{MAX}-\mathbf{MIN}})\cdot 60^{\circ},\ \ \text{if } \mathbf{B}=\mathbf{MAX}\end{cases}$$
$$\mathbf{S}=\frac{\mathbf{MAX}-\mathbf{MIN}}{\mathbf{MAX}} \tag{1}$$
$$\mathbf{V}=\mathbf{MAX}$$

where $\mathbf{MAX}$ and $\mathbf{MIN}$ are the maximum and minimum of RGB respectively. The value of the red bitmap at position x is extracted as follows:

$$\mathbf{RB}(x)=\begin{cases}1, & \mathbf{H}(x)\geq 2\pi-Th_1 \text{ or } \mathbf{H}(x)\leq Th_1, \text{ and } \mathbf{S}(x)\geq Th_2\\ 0, & \text{others}\end{cases} \tag{2}$$

where Th_1, Th_2 are two thresholds for hue and saturation components respectively. The extracted red bitmap is shown in Fig. 3(b).

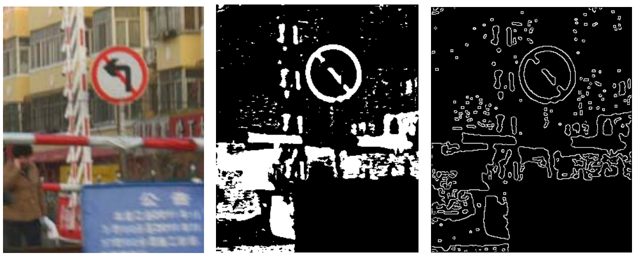

(a) (b) (c)

Figure 3. Extraction results of the red bitmap and the edge bitmap. (a) the input image (b) the red bitmap (c) the edge bitmap

B. *Edge bitmap extraction*

To perform circular Hough transform, an edge bitmap should be extracted from the red bitmap. Many edge detection methods [9, 10] can be used to extract edge bitmap. For computational simplicity, we choose the sobel operator. The extracted edge bitmap is shown in Fig. 3 (c).

C. *Circle detection*

We propose a circle detection method based on edge-following. The edge-following is a class of circle detection method which merges edge pixels into lines, arcs, and ellipses gradually. There are some circle detection methods [11, 12] based on edge-following are proposed, but the accuracy of which are still not satisfactory. We introduce a sorted merging strategy, which utilizes more global information and significantly increases the merging accuracy. The flow diagram of the proposed method is shown in Fig. 4.

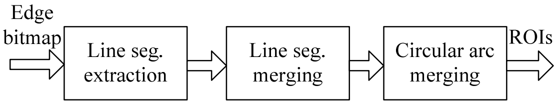


Figure 4. Flow diagram of the proposed circle detection method

Line segment extraction is the first step of the method, which aims at representing the edge bitmap with a series of line segments while preserving most of the information from the edge bitmap. Method in [13] is employed to complete this step and outputs a set $\{l_i\}$ of line segments. The extraction result is shown in Fig. 5(a).

After extracting all the line segments, line segment merging is performed to merge all the line segments into circular arcs. As the number of line segments is often up to several thousands, arbitrarily merging the line segments may easily lead to mis-merging. Therefore, we introduce a sorted merging strategy, which first estimating the merging degree of each pair of the line segments and then merge the line segments in the descending order of the merging degrees. The merging degree $D_{i,j}$ of two line segments l_i, l_j is defined as follows:

$$D_{i,j}=\frac{1}{d_{i,j}^{2}+1}\cdot\cos(\pi-\theta_{i,j})\cdot\frac{m_{i,j}}{M_{i,j}} \tag{3}$$

where $d_{i,j}$ and $\theta_{i,j}$ are the distance and intersection angle of the two line segments respectively, $m_{i,j}$ and $M_{i,j}$ are the minimum and maximum length of the two line segments respectively. With the equation above, the pairs of line segments with smaller $d_{i,j}$, larger $\theta_{i,j}$ and more similar lengths have larger merging degree, representing higher probability to be from the same circle.

After computing the $D_{i,j}$ of each pair of line segments, we start merging the line segments with the following steps:

(1) Initialize a set $\{a_i\}$ of circular arcs, each arc a_i contains a single line segment originally.

(2) Take out the pair of line segments l_i and l_j with the highest $D_{i,j}$ and have not been merged.

(3) Suppose the arcs corresponding to l_i and l_j are a_u and a_v respectively, fit an circle C with all the edge pixels in a_u and a_v and then calculate the fitting score of each edge pixel x with the following equation:

$$s(x) = \begin{cases} 1, & \min(\|x-y\|_2) \le 1 \quad \forall y \in C \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

(4) Get the fitting scores $\overline{S_u}$ and $\overline{S_v}$ of a_u and a_v by calculating the mean fitting scores of all the edge pixels contained in a_u and a_v. If $\overline{S_u} \ge 0.8$ and $\overline{S_v} \ge 0.8$, then merge the two arcs, otherwise do not merge.

(5) If all pairs have been processed, then stop the merging process, otherwise loop back to step (2) for the next pair of line segments.

After processing all the line segments, a merged set $\{a_i\}$ of arcs is generated, as shown in Fig. 5(b). It can be seen from Fig. 5(b) that some circles are split into multiple arcs, thus we need to further merge the circular arcs into circles.

Similar to the line segment merging process, we also first estimate the merging degree of each pair of circular arcs and then merge the arcs in the descending order of the merging degrees. The merging degree $F_{i,j}$ of two arcs a_i and a_j is defined as follows:

$$F_{i,j} = \min(\overline{S_i}, \overline{S_j}) \cdot \frac{\min[\text{len}(a_i), \text{len}(a_j)]}{\text{peri}(C_{i,j})} \tag{5}$$

where $\text{len}(a_i)$ is the length of a_i, $\text{peri}(C_{i,j})$ is the perimeter of circle $C_{i,j}$ fitted with a_i and a_j, S_i and S_j are the fitting scores of a_i and a_j to $C_{i,j}$. As the merging process is the same as line segment merging, we do not repeat it here. After circular arc merging, we get a set of circles, as shown in Fig. 5(c).

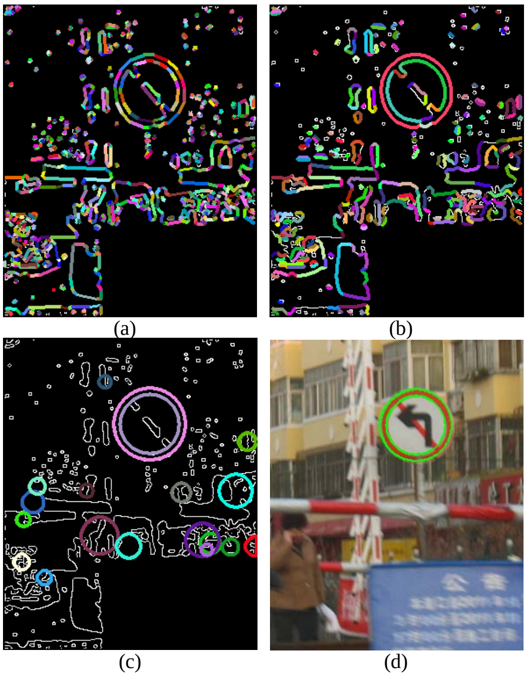


Figure 5. Illustration of the proposed detection method. (a) line segments (b) circular arcs (c) detected circles (d) detected circles after verification

D. *Verification*

As shown in Fig. 5(c), many circles are detected in the circle detection step, so we have to determine if the circles correspond to traffic signs. Since most of the pixels in a prohibitory sign are red, black or white, we can verify a candidate circle with the following criterion:

$$\begin{cases} N_r \ge 0.2N \\ N_b \ge 0.1N \\ N_w \ge 0.3N \\ N_r + N_b + N_w \ge 0.8N \end{cases} \tag{6}$$

where N_r, N_b, and N_w are the number of pixels in the circle with the color of red, black, and white respectively, and N is the number of all pixels in the circle.

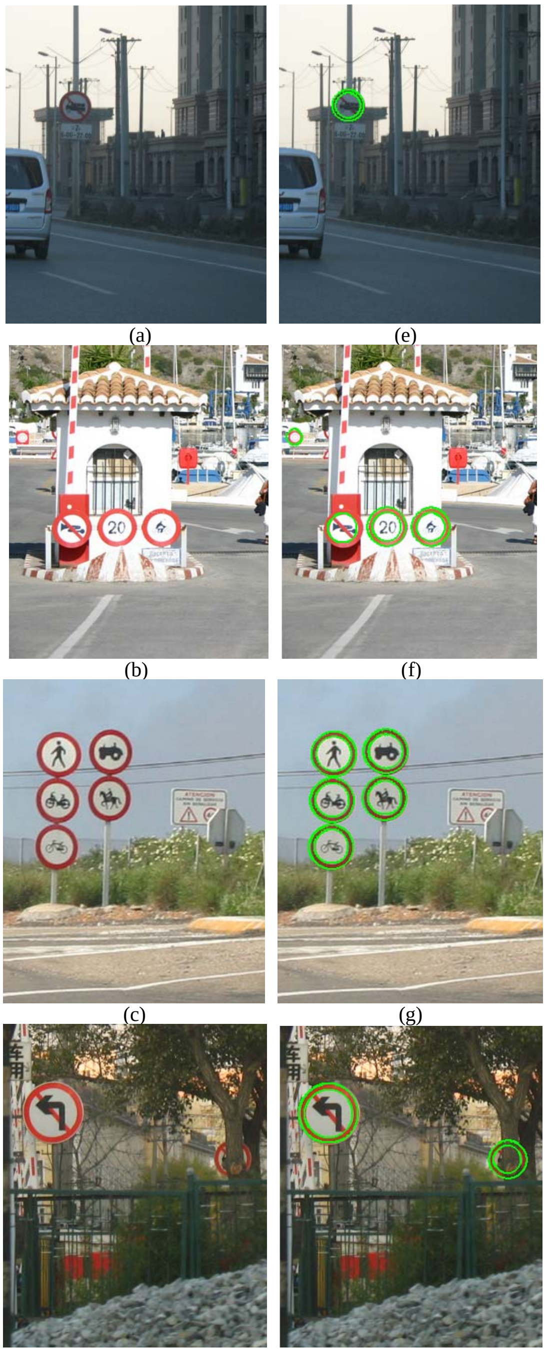
Circles satisfying the criterion are output as the final detection result, as shown in Fig. 5 (d).

III. EXPERIMENTAL RESULTS

In order to verify the accuracy of the proposed method, we test the method with a large set of images from real traffic scenes. Fig. 6 (a)-(d) are some representative examples, and Fig. 6 (e)-(h) are the corresponding detection results of them.

The detection results indicate that the proposed method is robust to all kinds of adverse situations including bad lighting condition, similar background color, multiple signs clustering, and partial occlusion.

Published by Atlantis Press, Paris, France.
© the authors



(a) (e)

(b) (f)

(c) (g)

(d) (h)

Figure 6. Detection results. (a) bad lighting condition (b) similar background color (c) multiple signs clustering (d) partial occlusion

The efficiency of the method is also tested. We implement the method with C++ and run the code on a Core I3 3.3 GHz CPU. The processing time for Fig. 6 (a)-(d) are listed in Table 1. It can be seen that the processing time changes with the number of red edge pixels in the test images. But in most cases, the method is fast enough for real-time applications.

TABLE I. PROCESSING TIME OF THE METHOD

	red edge pixels	Time (ms)
Fig. 6(a)	321	3.1
Fig. 6(b)	4619	71.6
Fig. 6(c)	3091	25.4
Fig. 6(d)	5143	77.2

IV. CONCLUSION

In this paper, we propose a new prohibitory traffic sign detection method. The method utilizes both the color and shape characteristics, which significantly improves the detection accuracy and efficiency. A new circle detection method is proposed which is fast and robust to noise and partial occlusion, while the computational complexity and memory consumption is much smaller than Hough transform. Experimental results show that the method can robustly detect prohibitory signs in all kinds of adverse situations and is efficient enough for real-time applications.

REFERENCES

[1] Stallkamp, J., M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition". Neural Networks. vol. 32, 2012, pp. 323-332.

[2] Fang, C.Y., S.W. Chen, and C.S. Fuh, "Road-sign detection and tracking". IEEE Transactions on Vehicular Technology. vol. 52, 2003, pp. 1329-1341.

[3] Maldonado-Bascon, S., S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-Sign Detection and Recognition Based on Support Vector Machines". IEEE Transactions on Intelligent Transportation Systems. vol. 8, 2007, pp. 264-278.

[4] Nguwi, Y.Y. and S.Y. Cho, "Emergent self-organizing feature map for recognizing road sign images". Neural Computing & Applications. vol. 19, 2010, pp. 601-615.

[5] Garcia-Garrido, M.A., M.A. Sotelo, and E. Martin-Gorostiza, "Fast road sign detection using hough transform for assisted driving of road vehicles", in Computer Aided Systems Theory. 2005, Springer Berlin, Heidelberg, pp. 543-548.

[6] Barnes, N., G. Loy, D. Shaw, and A. Robles-Kelly. "Regular polygon detection". in Proceedings of the Tenth IEEE International Conference on Computer Vision, 2005.

[7] Aoyagi, Y. and T. Asakura. "A study on traffic sign recognition in scene image using genetic algorithms and neural networks". in Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation, 1996.

[8] Escalera, A.d.l., J.M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles". Image and Vision Computing. vol. 21, 2003, pp. 247-258.

[9] Canny, J., "A Computational Approach to Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 8, 1986, pp. 679-698.

[10] Perona, P. and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion". IEEE Trans. Pattern Anal. Mach. Intell. vol. 12, 1990, pp. 629-639.

[11] McLaughlin, R.A., "Randomized Hough Transform: Improved ellipse detection with comparison". Pattern Recognition Letters. vol. 19, 1998, pp. 299-305.

[12] Fattal, R., "Single image dehazing". ACM Trans. Graph. vol. 27, 2008, pp. 1-9.

[13] Duda, R.O. and P.E. Hart, Pattern Classification and Scene Analysis. 1973, New York: Wiley.