

Generic Sign Board Detection in Images

Hua Shen

Department of Information Engineering

The Chinese University of Hong Kong
Shatin, Hong Kong

hshen1@ie.cuhk.edu.hk

Xiaoou Tang

Department of Information Engineering

The Chinese University of Hong Kong
Shatin, Hong Kong

xtang@ie.cuhk.edu.hk

ABSTRACT

Sign board detection is important for such computer vision applications as video surveillance and content based visual information retrieval. Previous researches on this topic focus mainly on application specific sign board such as car plates and traffic signs. Many special properties including special color, size, shape, and symmetry have to be used to detect these special sign boards. In this paper, we develop a system to detect generic sign board in an image or video. The only assumption we made is that the sign has to be a polygon formed by straight boundaries. Using a new set of straight line and polygon geometry analysis techniques, we can accurately locate a sign in a cluttered image efficiently. Experiments on over one hundred images clearly demonstrate the efficacy of the new algorithms.

Categories and Subject Descriptors

I.4.8 [Scene Analysis]: Object recognition, Shape.

General Terms

Algorithm, Design, Experimentation.

Keywords

Hough transform, Image analysis, Line verification, Sign board detection.

1. INTRODUCTION

Sign board detection is important for such computer vision applications as video surveillance and content based visual information retrieval. Previous researches on this topic focus mainly on application specific sign board such as car plates and traffic signs [1-7]. Color segmentation, gradient analysis, and Hough transform are generally used [1-5]. Some papers also used neural network or symmetry transform [6-7]. However, these methods are all limited by some application specific conditions, such as the shape of the car plates and the color of traffic signs.

In this paper, we present a system that can detect generic signs. The signs can be any polygon, not necessarily restricted to traffic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MIR '03, November 7, 2003, Berkeley, California, USA.

Copyright 2003 ACM 1-58113-778-8/03/00011...\$5.00.

signs or car plates. They can also be posters, name plates on doors, advertising board etc. The only assumption we made is that the sign has to be formed by straight boundaries. The system uses a gradient based Hough transform [8-10] to detect the boundary lines. We then develop a fast close circuit detection algorithm and a redundant line deletion technique to locate the sign board in the image. The algorithm is tested on over one hundred images.

2. PRE-PROCESSING

The pictures taken in outdoor environment are usually very cluttered. Fig. 1(a) shows such an example. So pre-processing is necessary to simplify the image before further processing. We propose three preprocessing steps for this purpose: edge detection, corner detection, and small segments deletion. For the first step, we use Canny edge detection algorithm. Figure 1 (b) shows the image after Canny edge detection.

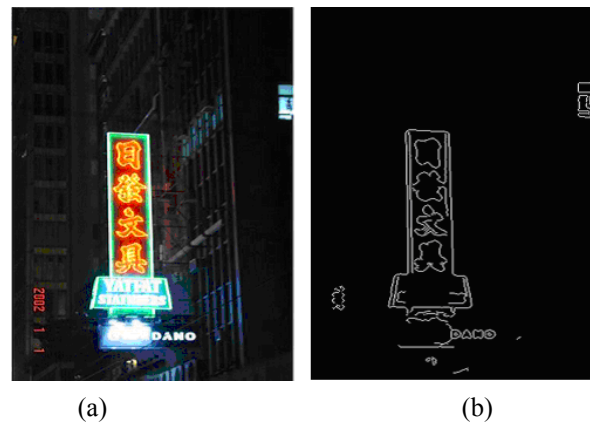


Figure 1. Original image and edges.

After edge detection, there are usually a large number of edges in a natural image, as shown in Fig. 2 (a). The signs are assumed to have relatively straight long boundaries. Other natural objects on the background may have randomly oriented interconnecting segments or curves as shown in Fig. 2 (a). We use a corner detection algorithm [11] to detect all the corners in the edge map, and then remove the corner points to break the interconnecting long edges into small segments. Since the corners do not affect long straight lines, we can therefore remove all the short segments broken by the corner points, as shown in Fig. 2 (b). Now the number of edges are much smaller thus enable a faster processing in the following steps.

3. GRADIENT BASED HOUGH TRANSFORM

Hough transform is an often used method to detect straight lines in an image [12]. But the main disadvantage of Hough transform is the large time and memory consumption, which is an important issue we consider for this application. So we use a significantly faster gradient based Hough transform instead [13].

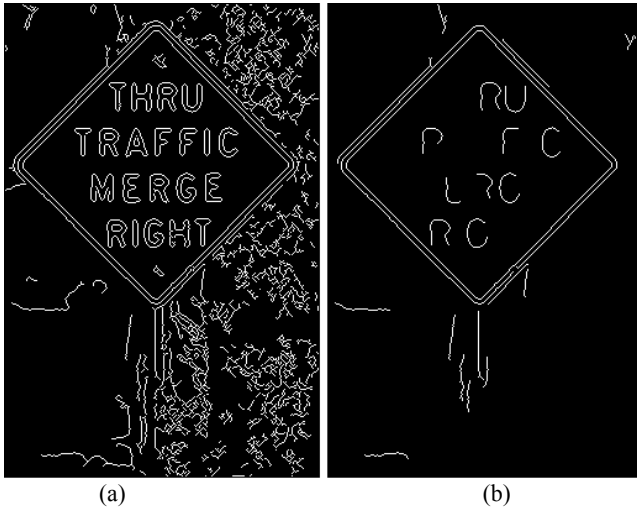


Figure 2. Illustration of cutting short segments.

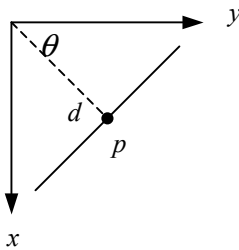


Figure 3. Parameters d and θ in Hough.

For Hough transform, a line function passes through an edge point p is determined by an angle θ and distance d : $x \cos \theta + y \sin \theta = d$, as shown in Fig. 3. Hough transform specifies all possible angles θ to the points, and calculates the corresponding distance d . By using an accumulator A for all d and θ , the peaks in A are the most probable lines in the image.

As mentioned above, Hough transform is very time consuming. So we use gradient based Hough transform instead. Notice that points are most likely to be on the line that has the same gradient as the point itself. So each point is calculated only for one angle θ and one distance d according to the gradient of the point. Although this algorithm is sometimes not very accurate due to the gradient computation error, it is still often used because of the great amount of time it saves. For the standard Hough transform, we need to consider the range of θ for $0^\circ - 180^\circ$ and $270^\circ -$

360° . Suppose the angle quantization interval is 5° . Then there will be $270/5=54$ angles for one point that we need to calculate the distance d . But if we use gradient information, we only need to calculate the distance once. The lines detected are shown in Fig. 4(b). The points in Fig. 4(a) are the edge points with the same gradient and distance as the lines in Fig. 4(b). They contribute to the peaks in the accumulator.

4. LINE VERIFICATION

After Hough transform, we get all the possible lines in the image. In order to find out the two ending points of each boundary segment, we compute the intersection points of all the lines first. Then the lines are cut into small segments by these intersections. And the corners of the signs should be among these intersections.

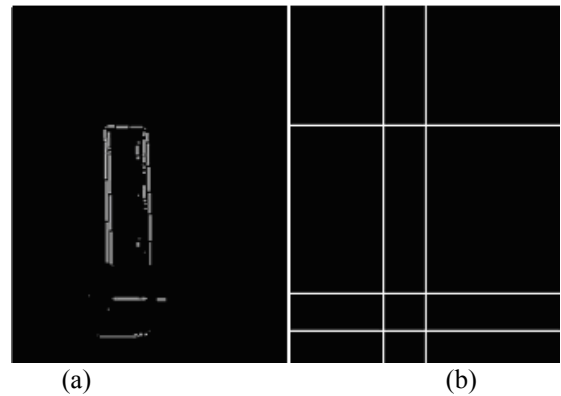


Figure 4. Lines detected and corresponding density points.

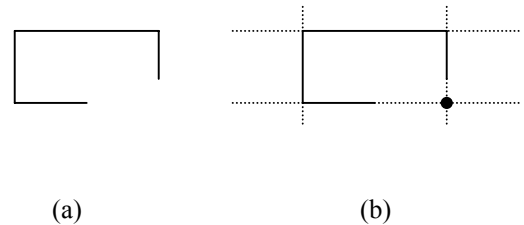


Figure 5. Intersection point that is occluded.

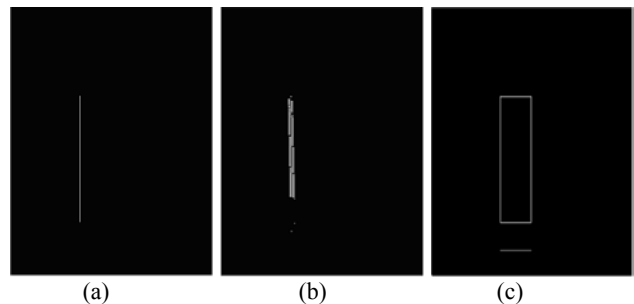


Figure 6. Lines before and after density checking.

If the segment is indeed the boundary, there should be many points with the same gradient and distance lying on that line. So we check the point's density on the segment then the segments with a density over a threshold are kept, and other segments are removed. For example, the points in Fig. 6(b) all belong to the segment in Fig. 6(a). Because of the high point density, this segment is kept as a boundary segment. Figure 6(c) shows the segments selected for lines in Fig. 4(b). The boundaries of the sign board are believed to be among these segments. Some more complicated results are shown in Fig. 7.

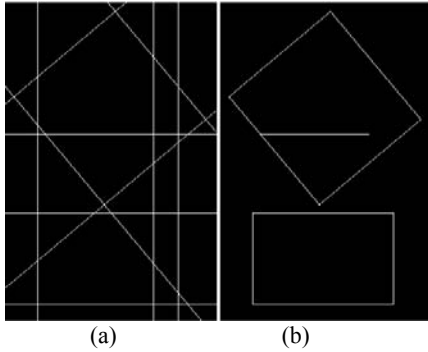


Figure 7. Line segment extraction results after checking point density.

4.1 Find Close Circuits

Now the candidate segments are found. We still need to decide which segments eventually form the boundaries of the sign. Since sign boundaries usually form a close circuit, we first detect all the close circuits among the line segments. Some of the segments may not be connected due to the noise, even though in fact they should be connected. So we first merge some intersections that are close enough. Examples of a set of closed segments are shown in Fig. 8(a).

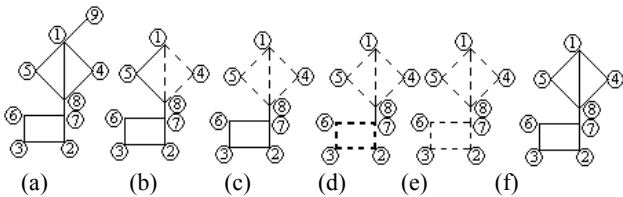


Figure 8. Steps of finding closed circuit.

Depth-first search is a popular method to find closed circuits [14]. If we want to find out every possible closed circuits, depth-first search need to be used for every points. The method is too time consuming for our application.

We propose a fast algorithm that first only finds one closed circuit in a connected graph. Then, each path with its two ends connected to that circuit is treated as an additional part of that closed circuit.

Repeat the process until the entire closed circuit is found. Using the example in Fig. 8(a), we illustrate the algorithm as follows.

First, delete the points with only one segment connected to it, like point 9 in Fig. 8(a). "Delete" here means delete the point and all the segments that end at this point. Double check until every point is connected with more than 1 segment.

Starting from the smallest number point (1) to find the first loop starts and ends at it. Number the loop as loop No.1, as shown by the dotted segments in Fig. 8(b). Put the loop's points (1,4,8) into a point set P. Put the path's segments (1-4, 4-8, 8-1) into a segment set S.

Then, try to find a path that starts and ends at the points in P, but the segments cannot be the ones in S. For example, path 1-5-8 satisfies this condition. Because its two ends 1 and 8 all belong to loop No.1, the path 1-5-8 also belong to path No.1, as shown by dotted segments in Fig. 8(c). Put the corresponding points (5) and segments (1-5, 5-8) into P and S.

Find every possible segment that can be added to the closed circuit No.1. Then start from the smallest point that is not in S. In this case, it is point (2). Again, try to find a closed loop. In the example, the loop is shown in bold dotted segments in Fig. 8(d). Because it is independent of loop No.1, number it as loop No.2. Also, put the points (2,3,6,7) into P and segments (2-3, 3-6, 6-7, 7-2) into S.

For the new points in P, we can find a path (7-8) that starts and ends at the points in P. But points 7 and 8 belong to different loop (No.1 and No.2). So we combine loop No. 2 into loop No. 1. Thus the loop No.1 becomes Fig. 8(e). Continue until all the points are checked. The final result is shown in Fig. 8(f).

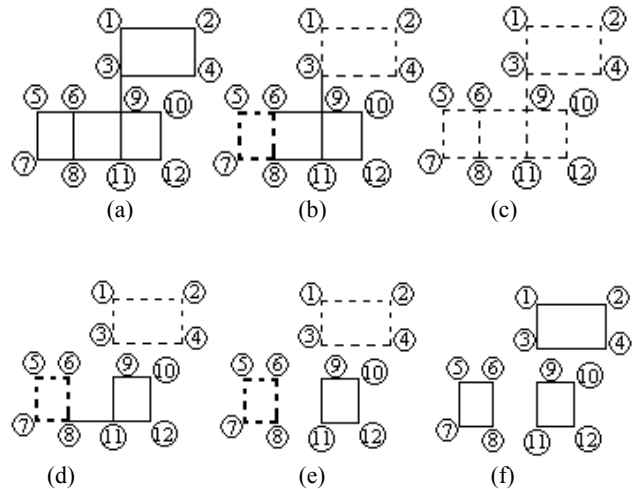


Figure 9. Examples of finding closed circuit.

The segment (7-8) seems redundant in Fig. 8. However, it should not be removed in the searching process. Because removing this type of segments may cause losing some loops, as shown by the example in Fig. 9. The dotted segments in Fig. 9(b) are the closed



Figure 14. Some example results.



Figure 15. Signs with corners covered in the image.

7. CONCLUSION

In this paper, we present a method to detect sign boards in image and video. The sign can be any polygon shape, and is not limited to rectangles or triangles. In our system, gradient based Hough transform is used. After the Hough transform, we compute all the line segments candidates by checking the point density of each segment. Then using a fast closed circuit detection algorithm and an efficient redundant line deletion method, the signs can be correctly detected. Experimental results on over one hundred images demonstrate the efficacy of the algorithm. In the future study, we intend to add texture information to the system. Since texture right inside the sign boundary tend to be smooth.

8. ACKNOWLEDGMENTS

The work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong SAR (Project no. CUHK 4357/02E and AoE/E-01/99).

9. REFERENCES

- [1] Eun Ryung Lee, Pyeoung Kee Kim, and Hong Joan Kim, "Automatic Recognition of A Car License Plate Using Color Image Processing".
- [2] N. Kehtarnavaz and A. Ahmad, "Traffic Sign Recognition in Noisy Outdoor Scenes".
- [3] Arturo de la Escalera, Miguel Angel Salichs, "Road Traffic Sign Detection and Classification", IEEE transactions on Industrial Electronics, Vol. 44, No. 6.
- [4] Paolo Comelli, Palo Ferragina, Mario Notturmo Granieri, and Flavio Stabile, "Optical Recognition of Motor Vehicle

License Plates", IEEE transactions on Vehicular Technology, Vol. 44, No. 4, November 1995.

- [5] Ming G. He, Alan L. Harvey, Thurai Vinay, "Hough Transform In Car Number Plate Skew Detection", ISSPA, 1996.
- [6] Yuji Aoyagi and Toshiyuki Asakura, "A Study on Traffic Sign Recognition in Scene Image Using Genetic Algorithms and Neural Networks."
- [7] Dong-Su Kim, Sung-II Chien, "Automatic Car License Plate Extraction Using Modified Generalized Symmetry Transform and Image Warping", ISIE 2001, Pusan, KOREA.
- [8] Heikki Kalviainen, Petri Hirvonen, Lei Xu and Erkki Oja, "Probabilistic and Non-probabilistic Hough Transforms: Overview and Comparisons", Image and Vision Computing Volume 13 Number 4 May 1995.
- [9] C. Galambos, J. Kittler and J. Matas, "Gradient Based Progressive Probabilistic Hough Transform", IEEE Proc. – Vis. Image Signal Process., Vol. 148, No. 3, June 2001.
- [10] J. Song, M. Cai, M.R.Lyu, S. Cai, "A New Approach for Line Recognition in Large-size Images Using Hough Transform", ICPR 2002.
- [11] Emanuele Trucco, Alessandro Verri, Introductory Techniques For 3-D Computer Vision, p82-85, Prentice Hall.
- [12] Hough, P. V. C., "A Method and Means for Recognizing Complex Patterns," U.S. Patent No. 3,069,654, 1962.
- [13] Robert M. Haralick, Linda G. Shapiro, Computer and Robot Vision, p578-582, Addison-Wesley Publishing Company, Inc. 1992.
- [14] William Ford, William Topp, Data Structures With C++, p748-751, Prentice Hall, 1996.