



ELSEVIER

Pattern Recognition Letters 17 (1996) 467-473

---

---

Pattern Recognition  
Letters

---

---

# A high-speed algorithm for line detection

Chun-Ta Ho, Ling-Hwei Chen \*

*Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC*

Received 6 March 1995; revised 7 December 1995

---

## Abstract

A high-speed method for line detection is proposed in this paper. By shifting the black points in a black/white image  $I$ , a parallel line for each straight line on  $I$  will be generated. Through the use of the geometric property on a pair of parallel lines, the parameter sets of those lines possibly on  $I$  can be obtained immediately. Since the transform from image space to parameter space is one to few points, the proposed method can significantly reduce the number of transforms for evaluating possible parameter sets. Experimental results are also given to show the correctness and effectiveness of the proposed method.

*Keywords:* Line detection; Parallel lines; Geometric property

---

## 1. Introduction

Detecting straight lines from an image is one of the basic tasks in computer vision. Hough transform for line extraction is well known and has been widely used (Hough, 1962; Duda and Hart, 1972). The method consists of three basic steps:

(1) transform a feature point in an input image into a curve on parameter space;

(2) each transformed point in parameter space is considered as a candidate and accumulated in the corresponding cell;

(3) a local peak-finding algorithm is applied to extract true parameters. These steps are feasible, but it causes the following two problems:

1. A lot of computing time is needed to transform each feature point in image space into a curve (many points) in parameter space.

2. For one feature point in image space, all possible set of parameters are accumulated in parameter space, but among them, only one set is correct.

Several modified versions of Hough transform have been proposed (Leavers, 1993). Some of them have realized that the computational load can be reduced using the coarse-to-fine accumulation technique (Li et al., 1986; Illingworth and Kittler, 1987; Atiquzzaman, 1992) or the random sampling concept (Xu et al., 1990). These methods try to solve the previously mentioned problems, but they still produce many superfluous points in parameter space. On the other hand, some use the local gradient information to reduce the dimension of lines (O'Gorman and Clowes, 1976; Davies, 1987), but they often suffer from poor consistency and locating accuracy due to noises and quantization errors in digital black/white images. Thus, developing a straight line extracting method, that is fast as well as easily applied to various practical applications, is very important. In the paper, we will present such a method.

---

\* Corresponding author.

First, the proposed method shifts each black point in a black/white image  $F$  with a certain distance. Through the shifting, a parallel line  $L_2$  for each straight line  $L_1$  on  $F$  will be generated. Then, for each black point  $A$  in  $F$ , by the geometric property on a pair of parallel lines, the parameters of those lines possibly passing through  $A$  will be obtained. Finally, based on the accumulative concept of Hough transform, each straight line on  $F$  will be extracted. Note that in the proposed method, for each black point, only few parameter sets are evaluated, and the local gradient information is not used.

## 2. The proposed method

For convenience, before describing the proposed method in detail, we will first introduce a definition and theorem, which will be used in the proposed method.

### 2.1. Definition and theorem

**Definition 1.** Let  $L_1$  and  $L_2$  be a pair of parallel lines,  $A(x_0, y_0)$  be a point on  $L_1$ . The *horizontal buddy point*  $B$  of  $A$  on  $L_2$  is defined to be the point on  $L_2$  with coordinates  $(x_1, y_0)$ . The *vertical buddy*

point  $C$  of  $A$  on  $L_2$  is defined to be the point on  $L_2$  with coordinates  $(x_0, y_1)$  (see Fig. 1).

**Lemma 1.** Let  $L_1$  be a line,  $L_2$  be the line formed by shifting each point on  $L_1$  toward right direction  $w$  pixels. Then  $L_1$  is parallel to  $L_2$ .

Lemma 1 is trivial, so the proof is omitted. Through Lemma 1, we can derive the following theorem.

**Theorem 1.** Let  $L_1$  be a line,  $L_2$  be the line formed by shifting each point on  $L_1$  toward right direction  $w$  pixels. Let  $A(x_0, y_0)$  be a point on  $L_1$  and  $C(x_0, y_1)$  be the vertical buddy point of  $A$  on  $L_2$ . Assume that  $L_1$  is expressed as (Duda and Hart, 1972):

$$\rho = x \cos \theta + y \sin \theta, \tag{1}$$

where  $\rho$  is the normal distance from the origin to  $L_1$ ,  $\theta$  is the angle between the normal of  $L_1$  and the  $X$ -axis. Then

$$\theta = \tan^{-1} \left( \frac{w}{y_1 - y_0} \right). \tag{2}$$

**Proof.** Through Lemma 1, we can see that  $L_1$  is parallel to  $L_2$ . Assume that  $B$  is the horizontal buddy point of  $A$  on  $L_2$ . Then  $B$  has coordinates  $(x_0 + w, y_0)$ . Since  $\overline{AC}$  is parallel to the  $Y$ -axis, from Fig. 1, we can get

$$\angle NPO = \angle CAP = \angle BCA.$$

Since  $\overline{OM}$  is the norm vector of  $L_1$ ,  $\triangle OMN$  is a right triangle. This leads to

$$\triangle OMN \approx \triangle PON,$$

and we can further get

$$\theta = \angle NOM = \angle NPO = \angle BCA.$$

Since  $\angle BCA = \tan^{-1}(w)/(y_1 - y_0)$ , this leads to

$$\theta = \tan^{-1} \left( \frac{w}{y_1 - y_0} \right). \quad \square$$

### 2.2. The proposed line detector

In this subsection, we will present the proposed method. Based on Theorem 1, the parameters  $(\theta, \rho)$  for each straight line on a black/white image  $F$  will be found. Initially, a two-dimensional array  $AR$  with

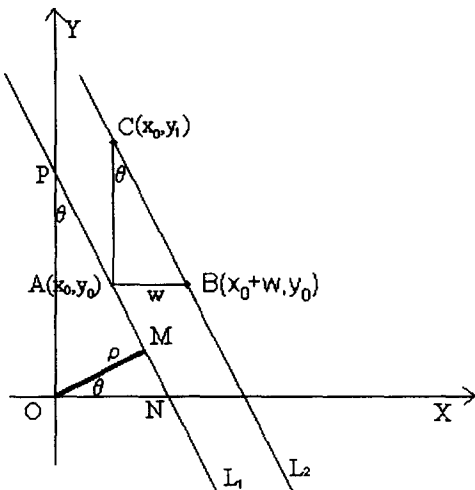


Fig. 1.  $L_1$  and  $L_2$  are a pair of parallel lines,  $L_2$  is generated by shifting  $L_1$  toward right direction  $w$  pixels.  $A$  is a point on  $L_1$ ,  $B$  and  $C$  are the horizontal and vertical buddy points of  $A$  on  $L_2$ , respectively.

$AR(\theta, \rho) = 0$  for each entry  $(\theta, \rho)$  is established. Next each black point  $D(x, y)$  in  $F$  is shifted toward right direction  $w$  pixels to get  $D'(x + w, y)$ , which will form another black/white image  $F'$ . Then for each black point  $A(x_0, y_0)$  in  $F$ , each black point  $C$  with coordinates  $(x_0, y_1)$  in  $F'$  is taken. By substituting  $(y_1 - y_0)$  and  $w$  into Eq. (2),  $\theta$  is first obtained, and then by putting the obtained  $\theta$  and  $(x_0, y_0)$  into Eq. (1), we can get  $\rho$ . The  $(\theta, \rho)$  are considered to be the parameters of a candidate straight line, and the corresponding entry  $AR(\theta, \rho)$  is increased by one. After all black points in  $F$  are processed, a local peak-finding algorithm is finally applied to array  $AR$  and each found peak is considered to be the parameters of a true straight line.

Note that if there exists a straight line  $L_1$  with parameters  $(\theta_0, \rho_0)$ , through the proposed method, a parallel line  $L_2$  will be generated. From Theorem 1, we can see that each point  $A(x_0, y_0)$  on  $L_1$  with its vertical buddy point  $C$  on  $L_2$  will contribute one to  $AR(\theta_0, \rho_0)$ . Hence, the value of  $AR(\theta_0, \rho_0)$  must be large and  $L_1$  can be found through the local peak-finding algorithm.

### 2.3. Applying the method to real cases

The proposed method can significantly reduce the number of transforms from image space to parameter space. However, applying the method to real cases still leads to two problems:

1. Noise or non-line black points cause many superfluous transforms.

2. Nearly vertical or horizontal lines can not be detected since correct vertical buddy points can not be located.

Based on the line property that if two points  $B$  and  $C$  are on a line, then their mid-point must be also on the line, a noise filter is added to treat Problem 1 and is described as follows.

In the proposed method, after  $A$  and  $C$  are located, the filter will take the mid-point  $M(x_m, y_m)$  of  $\overline{BC}$  as

$$M(x_m, y_m) = \left( x_0 + \frac{w}{2}, \frac{y_0 + y_1}{2} \right),$$

where  $B$  is the shifted point of  $A$ . Then  $M$  is checked to see whether  $M$  is a black point in  $F'$ . If

yes, point  $A$  is considered to be a point lying on a straight line and the corresponding parameter set  $(\theta, \rho)$  is computed and accumulated in the accumulation array. Otherwise, no more process is done.

On the other hand, in order to treat Problem 2, we apply the proposed method twice. First, the proposed method is applied to extract lines with  $|\theta| \in [22.5^\circ - 67.5^\circ]$ , then all black points are rotated clockwise  $45^\circ$  relative to the origin, and the proposed method is applied again to obtain lines with  $|\theta| \in [0^\circ - 22.5^\circ]$  or  $[67.5^\circ - 90^\circ]$ . The entire proposed method is described as follows.

### The proposed method

*Step 1.* Initialize a blank array  $AR$ ;

*Step 2.* Shift  $F$  toward right direction  $w$  pixels to get another image  $F'$ ;

*Step 3.* For each black point  $A(x_0, y_0)$  in  $F$   
For each black point  $C(x_0, y_1)$  in  $F'$   
Set  $M(x_m, y_m)$  to be

$$\left( x_0 + \frac{w}{2}, \frac{y_0 + y_1}{2} \right). \quad (3)$$

If  $M$  is a black point in  $F'$ , then

$$\theta = \tan^{-1} \left( \frac{w}{y_1 - y_0} \right). \quad (4)$$

If  $22.5^\circ < |\theta| < 67.5^\circ$ , then

$$\rho = x_0 \cos \theta + y_0 \sin \theta; \quad (5)$$

$$AR(\theta, \rho) = AR(\theta, \rho) + 1; \quad (6)$$

*Step 4.* The black points in  $F$  are rotated clockwise  $45^\circ$  relative to the origin as a new input image  $F_1$ ; Set  $F = F_1$ ;

Apply Steps 2–3, where Eq. (6) is modified as

$$AR(\theta - 45^\circ, \rho) = AR(\theta - 45^\circ, \rho) + 1;$$

*Step 5.* Apply a peak-finding algorithm to  $AR$ ;

For each extracted peak  $(\theta, \rho)$

Consider  $(\theta, \rho)$  to be the parameters of a true line.

### 3. The complexity analysis

In this section, we will analyze the computational complexities of the Hough transform and the pro-

posed method. For convenience of illustration, some symbols are defined as follows:

- $q$  = The number of intervals which span the parameter range when quantized;
- $n$  = the average number of black points on each column;
- $b$  = the average number of candidate buddy points for each black point;
- $t_c$  = the computation time of one comparison/shift operation;
- $t_a$  = the computation time of one addition/subtraction operation;
- $t_m$  = the computation time of one complex operation ( $*$ ,  $/$ ,  $\sin$ ,  $\cos$ ,  $\text{atan}$ ).

For the Hough transform, each black point is transformed into  $q$  sets of  $(\theta, \rho)$  through Eq. (1), the computation time needed is

$$qt_a + 2qt_m.$$

On the other hand, for each black point  $A$  and for each black point  $C$  on the same column as  $A$  in  $F'$ , the proposed method uses Eq. (3) to evaluate  $M$  and checks  $M$  to see if it is a black point, this will cost  $3nt_c + 2nt_a$ . Then, if  $M$  is a black point, a set of  $(\theta, \rho)$  is generated through Eqs. (4)–(5), this will need time  $2bt_a + 4bt_m$ . Thus, for each black point  $A$ , the total computation time is

$$3nt_c + 2(n + b)t_a + 4bt_m.$$

In practice, for most images (except some full of noise), both  $n$  and  $b$  are much smaller than  $q$  (this conclusion will also be shown in the experimental result, which will be described later),  $t_c$  and  $t_a$  are smaller than  $t_m$ . Thus, the proposed method will be faster than the Hough transform.

Note that in the proposed method, the shifting distance  $w$  will affect the preciseness of  $\theta$  and determine the minimum length of a line, which can be detected. The distortion of  $\theta$  can be estimated as follows. Let  $A(x_0, y_0)$  be a point on a line  $L_1$  with  $\theta = \theta_0$ . Let  $L_2$  be the line formed by shifting  $L_1$  toward right direction  $w$  pixels. Then the vertical buddy point  $C$  of  $A$  on  $L_2$  will have coordinates  $(x_0, y_1)$ ,  $y_1$  should be an integer due to digitization. However, the exact value of the  $y$ -coordinate of  $C$

should be  $y_0 + (w/\tan \theta_0)$ , and  $y_1$  is the nearest integer to  $y_0 + (w/\tan \theta_0)$ . So,  $y_1$  will be

$$y_0 + \left\lfloor \frac{w}{\tan \theta_0} \right\rfloor \quad \text{or} \quad y_0 + \left\lceil \frac{w}{\tan \theta_0} \right\rceil,$$

and the obtained  $\theta$  will be

$$\tan^{-1} \frac{w}{\lfloor w/\tan \theta_0 \rfloor} \quad \text{or} \quad \tan^{-1} \frac{w}{\lceil w/\tan \theta_0 \rceil},$$

the distortion for  $\theta_0$ , expressed as  $\Delta\theta_0$ , is

$$\Delta\theta_0 = \left| \tan^{-1} \frac{w}{\lfloor w/\tan \theta_0 \rfloor} - \theta_0 \right| \quad \text{or} \\ \times \left| \tan^{-1} \frac{w}{\lceil w/\tan \theta_0 \rceil} - \theta_0 \right|, \quad (7)$$

where  $|\theta_0| \in [22.5^\circ - 67.5^\circ]$ .

On the other hand, the minimum length of a line  $L_1$  with  $\theta = \theta_0$ , which can be detected, can be estimated as follows. Let  $A(x_0, y_0)$  be a point on line  $L_1$ ,  $C(x_0, y_1)$  be the vertical buddy point of  $A$  on the shifted line  $L_2$ . If  $C$  can be found, this means that the point  $C'(x_0 - w, y_1)$  is also on line  $L_1$  and on the input image. Thus the line  $L_1$  at least contains

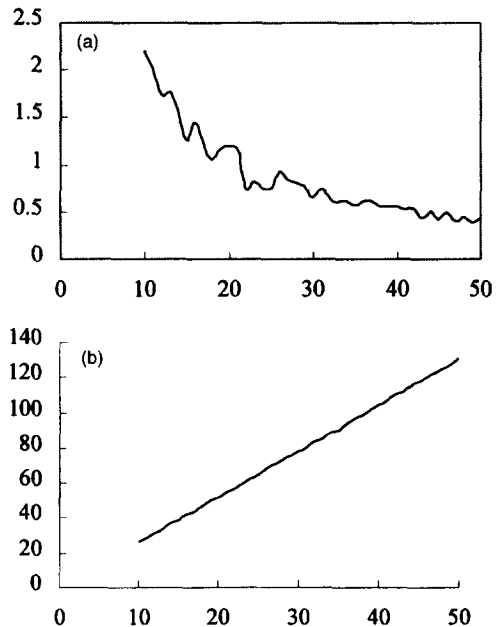


Fig. 2. The relationships of  $w$ ,  $\Delta\theta_m$  and  $l_m$ . (a) The relationship between  $w$  and  $\Delta\theta_m$ . (b) The relationship between  $w$  and  $l_m$ .

line segment  $\overline{AC}$ . Since  $y_1 - y_0 \cong w / \tan^{-1} \theta_0$ , the length of  $\overline{AC}$  is about

$$\sqrt{w^2 + \left(\frac{w}{\tan \theta_0}\right)^2}.$$

Thus, to make sure that line  $L_1$  can be detected, the minimum length of line  $L_1$ , expressed as  $l_{\theta_0}$ , should be larger than

$$\sqrt{w^2 + \left(\frac{w}{\tan \theta_0}\right)^2}, \quad (8)$$

where  $|\theta_0| \in [22.5^\circ - 67.5^\circ]$ .

Note that in our proposed method, lines with  $|\theta| \in [0^\circ - 22.5^\circ]$  or  $|\theta| \in [67.5^\circ - 90^\circ]$  are rotated clockwise  $45^\circ$  to make  $\theta$  lie within  $[22.5^\circ - 67.5^\circ]$ . Thus, for a line with such a  $\theta$ , the distortion  $\Delta\theta$  and

the minimum line length can also be evaluated through Eqs. (7)–(8). Define  $\Delta\theta_m = \max_{\theta} \Delta\theta$ ,  $l_m = \max_{\theta} l_{\theta}$ . Fig. 2 shows the relationships between  $w$  and  $\Delta\theta_m$  and between  $w$  and  $l_m$ , respectively. Referring to this figure, we can decide  $w$  based on the detecting minimum length and precision needed. In our experiment, for a  $512 \times 512$  image, we take  $w = 30$ .

#### 4. Experimental results

This section will present two experimental results of applying the proposed method to a  $512 \times 512$  real and a synthetic image. Fig. 3(a) is an image caught from an automatic land vehicle led by road-lines. Fig. 3(b) is the edge image of Fig. 3(a). Fig. 3(c)

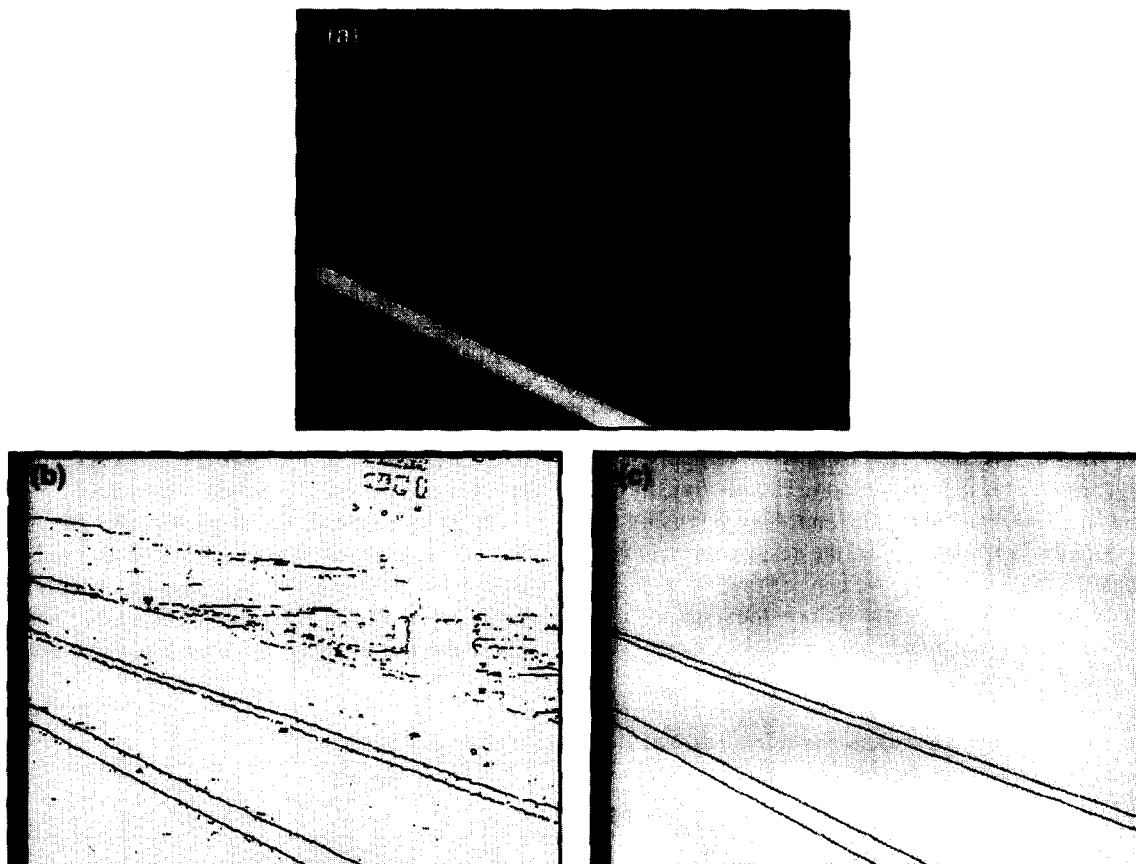


Fig. 3. The experimental result of applying the proposed line detector. (a) A real image. (b) The edge image of (a). (c) The detection result of applying the proposed method to (b); four straight lines are detected.

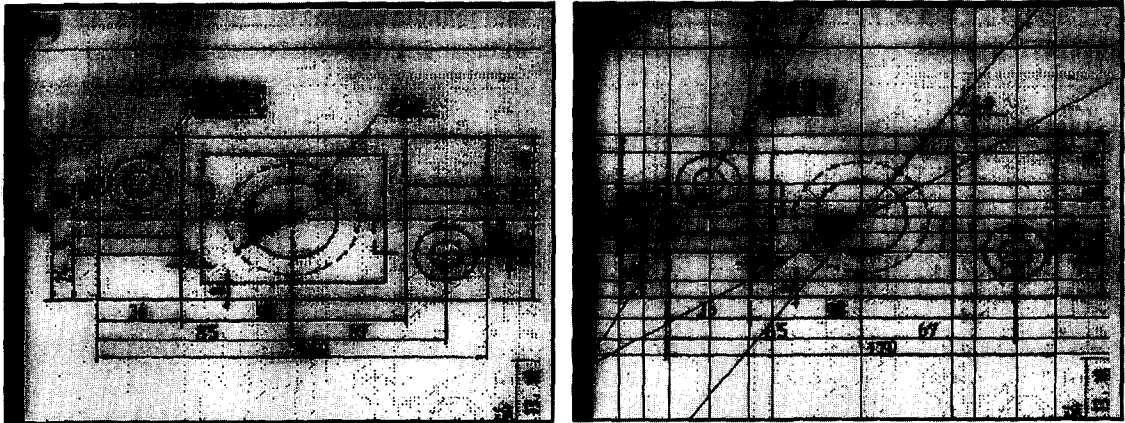


Fig. 4. The experimental result of applying the proposed method to a synthetic image. (a) A machine-part drawing. (b) The detection result superimposed on (a).

shows the detection result of applying the proposed method to Fig. 3(b) with shifting distance  $w = 30$ , four straight lines are detected. From this figure, we can see that the proposed method works well. Fig. 4 shows the experimental result of applying the proposed method to a machine-part drawing. From the detection result, we can see that most straight lines have been detected, only a few shorter lines have not. This is due to that a detected line must satisfy the following two conditions: (1) the length of the line must exceed the minimum length mentioned in Section 3; (2) the line must consist of enough points to get correct line parameters and to accumulate them in parameter space to form a peak. If a line cannot satisfy the above two conditions, the proposed method will fail to locate the line.

For the purpose of comparison, we also apply the Hough transform and the random Hough transform (Xu et al., 1990) to Figs. 3(b) and 4(a). By using a SUN SPARC-10 workstation, the results are shown in Table 1. From this table, we can see that  $b$ ,  $n \ll q$ , and our method can greatly reduce the number of transforms from the image space to the parameter

space and is faster than the other two methods. Note that there exists one problem in the random Hough transform, that is, how many random pairs should be taken. In our experiments, we use 15000 and 45000 pairs, respectively. These are try/error values.

#### 4. Conclusion

In this paper, we have proposed a new fast line detector. Based on the geometric property of a pair of parallel lines, for each feature point  $A$ , instead of evaluating all possible parameter sets in the parameter space, we only need to evaluate a few parameter sets corresponding to possible vertical buddy points of  $A$ . Hence, the proposed method is faster than the Hough transform. In addition, the proposed approach does not use the local gradient information.

On the other hand, the proposed method can also be used to detect bands. we just need to modify the method by adding a width parameter, which corresponds to the shifting distance  $w$ , for each band. Fig. 5(a) is the thresholded image of Fig. 3(a). For each

Table 1

The numbers of transforms and the running time needed by HT, RHT, and the proposed method

	Fig. 3(b) $q = 180, n = 9.1, b = 0.8$	Fig. 4(b) $q = 180, n = 36.3, b = 3.8$
HT	842220 (7.0 sec.)	3346380 (21.2 sec.)
Random HT	15000 (1.6 sec.)	45000 (5.1 sec.)
Proposed method	3923 (1.1 sec.)	71658 (4.5 sec.)

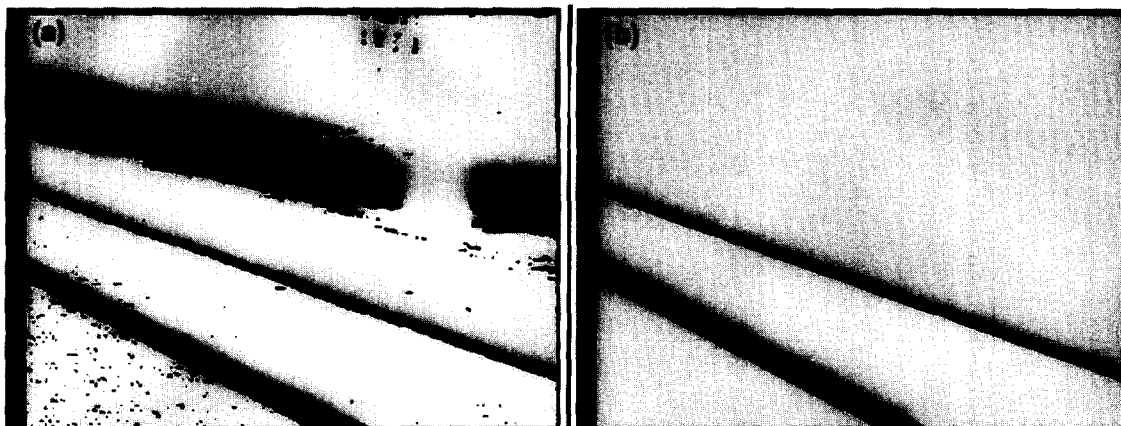


Fig. 5. The experimental result of applying the proposed band detector. (a) The thresholded image of Fig. 3(a). (b) The detection result of applying the proposed method to (a); two bands are detected.

edge point in the thresholded image, a set of buddy points is located, a set of possible parameters is computed and the corresponding entity is counted in a 3-d accumulation array. Finally, a peak-finding operation is applied to the array, the result is shown in Fig. 5(b).

### Acknowledgements

We would like to acknowledge the generous support by the National Science Council of R.O.C. under contract NSC-83-0408-E-009-037.

### References

- Atiquzzaman, M. (1992). Multiresolution Hough transform – an efficient method of detecting patterns in images. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 1090–1095.
- Davies, E.R. (1987). A new parameterisation of the straight line and its application for the optimal detection with straight edge. *Pattern Recognition Lett.* 6, 1–7.
- Duda, R.O. and P.E. Hart (1972). Use of Hough transformation to detect lines and curve in pictures. *Comm. ACM* 15, 11–15.
- Hough, P.V.C. (1962). Method and means for recognizing complex pattern. U.S. Patent No. 3069654.
- Illingworth, J., and J. Kittler (1987). The adaptive Hough transform. *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 690–698.
- Leavers, V.F. (1993). Survey: which Hough transform?. *CVGIP: Image Understanding* 58, 250–264.
- Li, H., M.A. Lavin and R.J. LeMaster (1986). Fast Hough transform: a hierarchical approach. *CVGIP* 36, 640–642.
- O’Gorman, F., and M.B. Clowes (1976). Finding picture edges through collinearity of feature points. *IEEE Trans. Comput.* 25, 449–456.
- Xu, L., E. Oja and P. Kultanen (1990). A new curve detection: random Hough transform. *Pattern Recognition Lett.* 11, 331–338.