A New Approach for Line Recognition in Large-size Images Using Hough Transform

Jiqiang Song, Min Cai, Michael R. Lyu, and Shijie Cai*
Dept. Computer Science & Engineering, the Chinese University of Hong Kong, Hong Kong, China
* State Key Lab. of Novel Software Technology, Nanjing University, Nanjing, China

Classical Hough transform based line recognition

Advantages:
- Robust in noisy environment
- Simple to implementation

Disadvantages:
- Very time-consuming for large size images
- Confusion between short lines and random alignments of pixels
- Unaware of line width

Our improvements:
- Boundary recorder
- Image-analysis-based line verification

Parameter cell
- Accumulator
- Low_boundary
- Up_boundary

Paradigm

Hough Transform & accumulation → Peak selection → Line verification → Final line results

Initialization: for all parameter cells

```
Param[r][θ].accumulator = 0;
Param[r][θ].low_boundary = max(X_MAX, Y_MAX);
Param[r][θ].up_boundary = min(X_MIN, Y_MIN);
```

Recording: when (x, y) contributes to (r, θ)

```
Param[r][θ].accumulator += 1;
IF (45°<θ<135°) {
  Param[r][θ].low_boundary=min(Param[r][θ].low_boundary, x);
  Param[r][θ].up_boundary=max(Param[r][θ].up_boundary, x);
}ELSE {
  Param[r][θ].low_boundary=min(Param[r][θ].low_boundary, y);
  Param[r][θ].up_boundary=max(Param[r][θ].up_boundary, y);
}
```

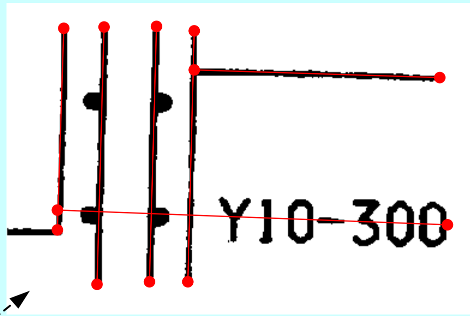

Figure 1. Line verification intervals

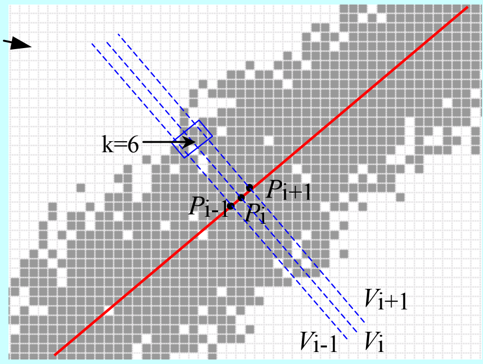

Figure 2. Local line width detection

Table 1. Performance over different image sizes

Image	A4	A3	A2	A1	A0
Size (Pixel²)	3533 ×2527	4990 ×3537	6959 ×4990	10078 ×6959	13783 ×10078
Storage (MB)	1.05	2.11	4.18	8.41	16.78
Time (sec.)	17.4	39.3	81.5	152.7	303.6
T_LV (sec.)	3.8	8.9	18.3	41.6	87.1
LS_Num	547	1018	1778	3240	5218
T/LS (msec.)	6.95	8.74	10.29	12.84	16.69
Recog_Rate	0.89	0.90	0.91	0.88	0.87

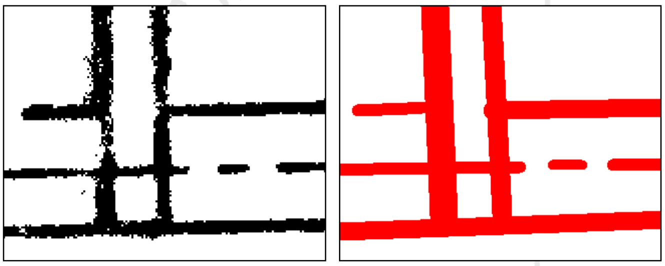

Figure 3. Line recognition result of a real image
(The detected line width is displayed.)

- Make Hough transform applicable to large images.
- Enable HT-based line recognition to detect line width.

Notes:

Hough transform has been employed for line recognition for a long time due to its anti-noise ability and simplicity. However, it has three disadvantages: first, for large size images, it is very time-consuming; second, random alignments of pixels also generate false peaks, which mix up with short lines. Third, it cannot detect line width. Our approach is to solve these problems.

The paradigm of Hough transform based line recognition consists of three steps.

(1) Hough transform & accumulation.
(2) Peak selection to get the parameters (r, θ) of possible lines.
(3) Line verification: for each parameter, verify the real connectivity of those feature points contributing to this parameter.

Many variations of Hough transform have accelerated the first two steps to a large extent. Thus, when applying the Hough transform based line recognition to large images, the line verification becomes the time-consuming part, which could be accelerated. The other two disadvantages are also related to the line verification step. However, the improvement on this step is seldom addressed. Therefore, we focus on this step and make two improvements. Using the boundary recorder in the accumulation step can speed up the line verification step, and the image-analysis-based line verification can distinguish short lines from the random alignments of pixel and detect line width as well.

In the image space, a selected peak indicates a strip area along the line direction. One important reason for the time-inefficiency of other line verification methods is that they do not know which part of the strip area contains feature points. Usually, only a small part of the strip area contains the feature points. According to this fact, we add a boundary recorder to each parameter cell (as the left illustration), which only contains an accumulator before, to record the minimum scope that contains the feature points contributed to this parameter. During the accumulation, this code will calculate the up boundary and the low boundary of the feature points contributing to each parameter. Thus, only the interesting part of the strip area is analyzed when verifying the line.

After selecting peaks by a threshold, we perform the line verification by analyzing the image along the line direction. To avoid the heavy computation in solving the line equation, we use a rasterization method to generate the straight-line path between two boundaries. The black pixel segments along the path are analyzed. Based on the image information, we can easily distinguish between real short lines and the random alignments of pixels (as shown in Fig. 1). Dashed line styles can also be detected. The line width is voted by all local line widths detected at each black pixel on the line path. Fig. 2 shows the approach to detect local line width, which can tolerate the poor quality of image.

After all line segments contributed to a peak have been verified, the pixels of these line segments should be removed from the image to avoid overlapping lines.

Figure 3 shows the recognition result of a real image. The degraded lines are correctly recognized with accurate line widths. Table 1 shows the performance over images of different sizes. We find both the speed and the recognition rate are satisfactory. Note that the time for line verification (T_LV) is only a small portion of the whole processing time (Time). T/LS = T_LV / LS_Num.

In summary, this paper has two contributions:

(1) Make Hough transform applicable to large images.
(2) Enable Hough transform based line recognition to detect line width.