Form Frame Line Detection with Directional Single-Connected Chain

Yefeng Zheng, Changsong Liu, Xiaoqing Ding, Shiyan Pan

Department of Electronic Engineering, Tsinghua University
Beijing, 100084, P. R. China
Email: zhengyf@hotmail.com

Abstract

In this paper, a novel form frame line detection algorithm is proposed based on the Directional Single-Connected Chain (DSCC). Defined as an array of black pixel run-lengths, DSCC works very well as an image structure element or vector in our vectorization algorithm. By merging multiple DSCCs under some constraints, we are able to extract the form frame lines automatically yet fast. The speed of our algorithm is comparable with some well-known projection methods. Experiments show that our algorithm is fast, resistant to moderate serious line breaks and can detect diagonal lines with any angle.

1. Introduction

Form is widely used to collect and distribute data in daily office operations. As a high constructed document, form consists of characters and some structured horizontal, vertical, and diagonal frame lines. Frame line detection is the most important and difficult step of form recognition. Hough transform [1] and vectorization [4] are two kinds of widely used line det ection methods. As a global approach, Hough transform can detect dashed or broken lines. However, it is too slow to be applied in form recognition. Fortunately, most frame lines on forms are horizontal or vertical. So in the (ρ,θ) transformed space, we can narrow the search range of θ to small areas around 0° and 90°. Such modified Hough transform is just projection approaches [2,6]. Though fast, projection approaches have some problems. First, they cannot detect diagonal lines and frame lines with large skew angles. Second, when characters overlap frame lines, the projection of frame lines are overwhelmed in the projection of characters. Such lines cannot be detected correctly. Third, some frame lines in a scanned image, especially those on the image borders, are deformed. With some kind of cursive, they are not straight now. Projection methods fail to detect such curved lines too. As the other kind of algorithms widely used, vectorization approaches [4] extract vectors from images first. By merging these vectors, the whole objects are detected. Such bottom-to-up approaches can solve the above problems of projection approaches.

In this paper, we presented a kind of vectorization algorithm, which uses a novel image structure element named "Directional Single -Connected Chain (DSCC)"as the elementary vector. DSCC bears appropriate size and can be easily stored and processed, in addition to the capability to solve most types of character-line crossing problems. By merging DSCCs under some constraints, most of the frame lines can be detected correctly. However, there may still exist two kinds of mis detection, i.e., the pseudo lines and the broken lines. Fortunately, frame lines in a form must have some constraints to compose meaningful form cells, which can be utilized to reduce misdetection. Those lines, which are not used to compose form cells, are pseudo-lines and removed. Some seriously broken line segments are also merged together with the help of form frame line constraints. Generally, vectorization approaches are much slower than projection approaches due to the large number of vectors. We exploited some effective methods to accelerate our algorithm. After speeding-up, the speed of our algorithm is much faster than the Hough transform,

0-7695-1263-1/01/$10.00 © 2001 IEEE

and comparable with some well-known projection algorithms [2,6].

2. Definition of Directional Single-Connected Chain (DSCC)

We defined two kinds of directional single-connected chains: the horizontal single-connected chain and the vertical single-connected chain, which are used to detect horizontal and vertical frame lines respectively. Diagonal lines can be detected with either horizontal DSCC or horizontal DSCC. Taking the horizontal DSCC for example: horizontal DSCC C_h is made up of black pixel run-length array $\overrightarrow{R_1R_2\cdots R_m}$, where each R_i is one pixel width vertical run-length.

$$R_i(x_i, ys_i, ye_i)=\left\{(x,y)\,\middle|\,\begin{matrix}\forall p(x,y)=1, x=x_i, y\in[ys_i, ye_i]\\ p(x_i, ys_i-1)=p(x_i, ye_i+1)=0\end{matrix}\right\} \quad (1)$$

$p(x,y)$ is the value of (x,y) pixel, 1 represents black pixels, 0 represents white pixels; x_i, ys_i and ye_i designate the x coordinate, starting y coordinate and ending y coordinate of R_i respectively. Each two neighboring run-length R_i and R_{i+1} meet the following conditions:

1) Connected in horizontal direction: $x_{i+1}=x_i+1$
2) Single connected.

As show in figure 1, single connection means that on each side of R_i (i?1 and i?m), there is one and only one connected run-length belonging to C_h. It is true to the right side of R_1 and the left side of R_m too. But on the left side of R_1 and the right side of R_m, there are no connected run-lengths (e.g. the right side of R_{13}), or more than one connected run-lengths (e.g. R_{15} and R_{14} on the left side of R_1), or though there is only one connected run-length, but this run-length is connected to more than one run-lengths on the same side of R_1 or R_m(e.g. R_9).

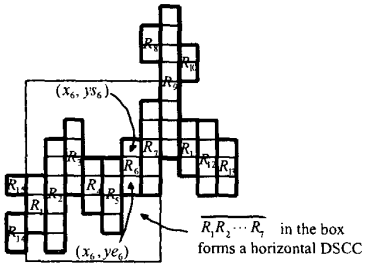


Figure 1. Horizontal DSCC

The definition of vertical single-connected chain C_v is very similar to the horizontal single-connected chain. We do not describe it here in details.

The most important property of a line is single connection along its running direction. A ideal line is made up of only one DSCC; While a real line, which is broken or crossed by characters, is made up of more than one un-overlapped DSCCs. Making full use of this property, DSCC is very convenient to detect straight lines.

3. Form Frame Line Detection with DSCC

3.1 DSCC Mergence

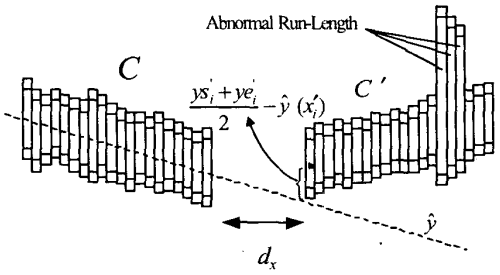


Figure 2. Co-line distance

Every frame line is made up of one or more un-overlapped DSCCs. By merging such DSCCs, we can get the whole frame line. "Co -line Distance", which indicates the possibility of two chains to lie on a line, is defined to select an appropriate chain to be merged.

Taking the horizontal chain for example: supposing the mid-point fitting of a horizontal DSCC $C=\overrightarrow{R_1R_2\cdots R_n}$ is $\hat{y}(x)$, the co-line distance of the other chain $C'=\overrightarrow{R_1'R_2'\cdots R_m'}$ to C (as shown in Figure 2) is:

$$d_{C'C}=\begin{cases}\infty, & when\ d_x\le 0,\\ d_x+\dfrac{\sum_{\substack{i=1\\ R_i'\notin H}}^{m}\left(\frac{ys_i'+ye_i'}{2}-\hat{y}(x_i')\right)^2}{M} & when\ d_x>0\end{cases} \quad (2)$$

$$d_x=\max(x_1, x_1')-\min(x_n, x_m') \quad (3)$$

x_1 and x_n are the x coordinates of the left and right edges of chain C respectively. x_1' and x_m' are the x coordinates of the left and right edges of chain C'. As showed in figure 2, d_x is the horizontal distance of C and C'. If d_x=0, C and C' overlap in the vertical direction. It is impossible for them to lie on a line, and $d_{cc'}$ is set to infinite. The second term of $d_{cc'}$ indicates the distance from mid-points of C' to the extended line of C. The smaller the value, the greater the possibility for C and C' to lie on a line. Only "normal run -length", whose length is smaller than two times of the average width of the chain, is used to calculate co-line distance. The effect of "abnormal run-length" is restrained. M in the equation (3) is the number of normal run-length of chain C'.

C' can be merged with C only if it satisfies the following conditions:

1). Linear extending condition: $\sqrt{d_{C'C} - d_x} < W$, W is the average width of C.

2). Gap condition: In the area between C and C', there are three possible cases:

(a) Empty: If d_x=T_1 (T_1=15), C and C' are merged. Otherwise the gap is too large, C and C' cannot be merged.

(b) Exist another chain, and its width is less than two times of the average width of C: It is treated the same as the first case.

(c) Exist another chain, and its width is larger than two times of the average width of chain C. Here the horizontal line is interrupted by a vertical line. A smaller threshold T_2 (T_2=8) is used. If d_x=T_2, merge C and C', else do not merge them.

The steps of mergence is:

1) The longest chain not processed yet is selected as a seed chain C_s

2) On one side of C_s, sort chains C_i' (i=1,2...n) with the co-line distance ascending.

3) Among the first M (M=3) chains, if we can find chain C_k' with the smallest co-line distance and fill the two merging conditions listed above, then merge C_s and C_k'.

4) Repeat the step 2 and 3 on two sides of C_s

5) If all chains are processed then end mergence, else go to step 1.

3.2 Estimation of Character Size:

In most frame line detection algorithms, a critical threshold is used to remove any short lines formed by character strokes. This threshold represents the character size. However due to the difference of forms and resolution of scanners, the width or height of characters varies greatly from 10 pixels to more than 100 pixels. In most literature, this important threshold is input by users [3] or is a constant value [2, 6]. We proposed a method to estimate the size of characters automatically using connected component analysis based on run-lengths. Two histograms are formed: the width histogram and the height histogram of the connected components. In a form with only one font size, there is only one peak on the histogram, which is the width or height of characters. But in most cases, characters with different font sizes appear in a form at the same time. There will be more than one peaks on the histograms. The largest peak, which corresponding to the largest font used, is selected as the size of characters. Then, we remove all horizontal lines shorter than the character width and all vertical lines shorter than the character height, which are formed by strokes of characters.

3.3 Pseudo Line Removal and Broken Line Completion

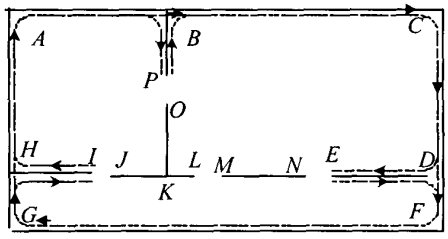


Figure 3. Three form cells merged because of line breaks

After mergence, most lines are extracted correctly. But there still exist two kinds of errors. One kind is pseudo lines, which is formed by some strokes lying on a line occasionally. This problem is very serious in Chinese forms because most Chinese character strokes are horizontal or vertical. The other errors are line breaks.

Some lines are broken so serious that we cannot link them up without the help of constraints between form frame lines. In order to use such constraints, form cells are formed [3]. Because of line breaks, some form cells may be merged as a single cell, as show in figure 3. Some segments inside the cell lie on a line. If they are long enough (for example longer than 0.8 of the cell width or height), we link them up and split the cell. Continue such split until no segments meet the split conditions. After introducing form frame line constraints, all lines not used to compose form cells are removed as pseudo lines, and some serious broken lines are completed.

4. Speeding Up of the Algorithm

Speed is an important factor to evaluate a form processing system. Generally, vectorization algorithms are much slower than projection algorithms because of large vector number. The DSCC number of a typical form with dimension 1000*1000 pixels is about 40,000 to 60,000. In this section, we discuss some methods exploited to accelerate our algorithm, and while preserving the robustness of the system at the same time.

4.1 Run-Length Smearing

Due to printing, copying or scanning, there are some small breaks in frame lines. By merging those run-lengths with gaps smaller than a threshold, the run-length number is reduced greatly. Experiments show after run-length smearing, the number of DSCC reduces 20%, and the robustness to line breaks increases too.

4.2 Removal of Small DSCC

More than 50% of the DSCCs are made up of chains less than 3 pixels in length. Most of these small chains come from characters and noise, very few of them come from broken lines or dashed lines. Chains fitting either of the following conditions are removed:

1) No more than 2 pixels in length (representing chains coming from noise).
2) No more than 5 pixels in length and having connected chains on one side or both sides (representing chains coming from characters).

Because the chains formed by broken lines and dashed lines are not connected with other chains and longer than 2 pixels, most of them are preserved. For a typical form image, about half of DSCCs are removed.

4.3 Reduce Searching Areas During Mergence

DSCC mergence takes up most of the processing time. Supposing the number of DSCC is proportional to area, the computation complexion approximates to:

$$(k_1 \times W \times H) \times (k_2 \times W \times H) \quad (4)$$

W is the width of the form, H is height of the form, k_1, k_2 are constants. The former term is the DSCC number; the latter term is the number of DSCCs searched each step during mergence. We split the image into several strips with the same size. Lines are extracted in each strip independently. Then the results of each strip are combined. For example, the whole image is divided into several horizontal strips with same height to detect horizontal lines. While during vertical line extraction, the whole image is divided into several vertical strips. After division, taking horizontal line extraction for example, the computation complexion approximates to:

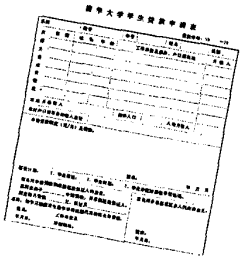
$$(k_1 \times k_2 \times H^2 \times W^2) \times (w \div W) \quad (5)$$

Ignoring the overhead of combination, if W=2000 and w=400, the speed increases up to 5 times.

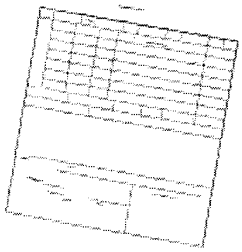
5. Experiments

In the first experiment, we test the robustness of our approach to skewness and line breaks. The form image is rotated 10°. As show in figure 4-(b), after DSCC mergence, there are some pseudo lines and some lines are broken. After using the constraints of frame lines, as show in figure 4-(c), all form frame lines are extracted correctly. We tested our approach on a form image database containing 200 forms. Among them, 150 forms are with good quality and 50 forms are with serious line breaks. The average line detection rate on the former is

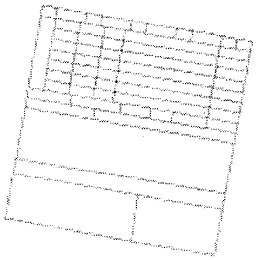
over 98%, and the correct rate on the latter is about 93%.



(a)



(b)



(c)

Figure 4. (a) Form image. (b) DSCC mergence result. (c) Lines extracted after pseudo-line removal and broken line completion.

In the next experiment, we test the effect of the speeding up methods and compare the speed of our approach with some well-known projection methods: skew projection [2] and strip projection [6]. Test condition: Pentium II 233MHZ CPU, with 64M memories. The result is listed in table 1.

Table 1. Speed comparison

Image Size (Pixels*Pixels)	1	2	3	4	Speed Increased
684*650	0.3s	0.5s	0.8s	0.6s	1.3
1816*1112	0.5s	0.8s	2.9s	1.0s	2.9
2400*3438	0.8s	1.6s	13.5s	1.9s	7.1
4198*3165	1.5s	3.8s	109s	5.0s	21.8

1. Skew projection 2. Strip projection

3. DSCC before speeding up 4. DSCC after speeding up

To ordinary sized forms, the speed can increase 3 to 10 times. To form image 4, which is large, with high black pixel ratio and a lot of noise generated during scanning, the speed increased more than 20 times. After speeding up, the speed of our approach is comparable with the projection methods.

6. Conclusions

We proposed and realized a novel form frame line detection algorithm based on Directional Single Connected Chain. After chain mergence, form frame line constraints are used to remove pseudo lines and link serious broken lines up. Experiments show that our approach can extract diagonal lines with any angle, and is resistant to moderate serious line breaks. Our algorithm can extend to gray images. Using a self-adaptive "valley"detection algorithm, we can build DSCC directly on gray images.

References:

[1] J. Illingworth and J. Kittler, "A Survey of the Hough Transform", *Computer Vision, Graphics, & Image Processing*, vol.44, 1988, pp.87-116

[2] Jinhui Liu, Xiaoqing Ding, Youshou Wu, "Description and Recognition of Form and Automated Form Data Entry", In *Proceedings of 3rd ICDAR*, Montreal, Canada, 1995, pp. 579-582

[3] Shiyan Pan, "Rese arch and Realization of a General Form Recognition System", M aster thesis of Tsinghua University, June, 1999

[4] Wenyin Liu, Dov Dori, "From Raster to Vectors: Extracting Visual Information from Line Drawings", *Pattern Analysis & Application*, No.2, 1999, pp.10-21

[5] Bin Yu, Anil K. Jain, "A Generic System for Form Dropout", *IEEE Trans. On Pattern Analysis & Machine Intelligence*, Vol.18, No.11, 1996, pp.1127-1131

[6] Jiun-Lin Chen, Hsi-Jian Lee, "An Efficient Algorithm for Form Structure Extraction Using Strip Projection", *Pattern Recognition*, Vol.31, No.9, 1998, pp.1353-1368