Design of Text-image Separation Algorithm Based on FIR filter and Its Application in Photocopying Equipment

Chunyan Tang, Xijing Guo and Jie Gao

Institute of public basic and applied statistics, Zhuhai College of Jilin University, Zhuhai 519041, China

***Abstract*—The traditional image processing method has the problem that the system running time is too long, or the hardware resource of the photocopy equipment is high. This paper analyzes the features of the page data of the dot area and text line area in the hard copy manuscript. Based on the difference of these features and the principle of FIR filtering, this paper presents a text-image separation algorithm for photocopy equipment. The algorithm takes the gray data as input, realizes the complicated text-image separation and processing process through the gradient detection and the region transformation, and reduces the hardware resource requirement greatly. The experimental results show that the algorithm is accurate and efficient, and has good practicability.**

Keywords—photocopying equipment; text-image separation; FIR filtering; dot

I. Introduction

With the popularization of all-in-one printers (MFP) and other photocopying equipment, users not only hope that the equipment performance is good, the price is cheap, but also the demand for image quality is more and more high. They always want to get text sharp and smooth image in the printed manuscript. Therefore, it is necessary to design an image processing algorithm which satisfies the good performance, low cost and high quality of the equipment.

Users often use printed manuscripts when scanning and copying. The image portion of this manuscript is made up of a number of black and white dots of different sizes due to limitations of the printing process. This type of manuscript is limited by the printing or printing process, and the image is made up of several black and white dots of varying sizes. So that images and text cannot be enhanced simultaneously. In addition, because the edge features of text and lines are very similar to dot regions, simple threshold segmentation algorithms are difficult to distinguish between the two. Therefore, in order to obtain better image quality, it is necessary to separate the text and image effectively.

With the research of some institutions, a variety of image separation algorithms have been proposed [1~6]. These algorithms solve some specific problems more or less, but they either have a long processing time or require high device hardware resource. Table 1 lists the performance parameters of two main text-image separation algorithms in the common case of A4 format paper and 300*300 dpi configuration.

TABLE I. PERFORMANCE COMPARISON OF MAINSTREAM TEXT-IMAGE SEPARATION ALGORITHMS

Algorithm name	**Input data type**	**ART^a (s)**	**MER^b (MB)**	**Image processing steps (black-and-white output)**
Multi-convolution algorithm	Color or gray	5.72	8.3	Color or grayscale input -> text-image separation -> binary output
Color separation algorithm	Color	4.83	24.9	Color input -> text-image separation -> color space conversion -> binary output

a. Average run time

b. Minimum memory required

Note: If MFP speed is 25 ppm, i.e. one minute output 25 pages, the time required to process each page should be < 60/25 = 2.4 seconds.

From table 1:

(1) Although the multi-convolution algorithm requires low memory and simple image processing, its average running time is longer. In the actual application will cause the office equipment page output slows, affects the user experience.

(2) Although the color separation algorithm has the advantage in average running time, the memory requirement is 3 times of the multi-convolution algorithm. Additionally, additional processing steps are required in the monochrome output device, resulting in increased hardware costs and reduced real usage.

Therefore, it is of great practical significance to design a text-image separation algorithm with less requirement for hardware resources and short processing time. It not only improves image quality and reduces machine costs, but also improves user experience.

II. Analysis of Printing Data Characteristics

At present, the common output resolution in the printing field is 133 line/inch and 200 line/inch. The former is used in ordinary print, such as newspaper. The latter applies to high-grade prints, such as magazines. The two types of output resolution corresponding to the dot distribution characteristics and the scanning output diagram are shown in Figure 1 and Figure 2 respectively.

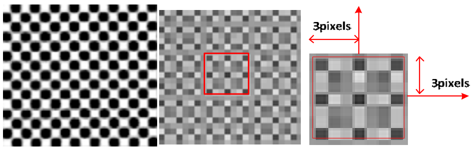

FIGURE I. 133 LINE/INCH COMPARISON AND ANALYSIS OF DOT IMAGE SCANNING BEFORE AND AFTER

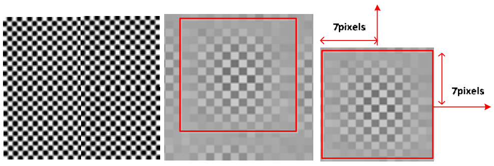

FIGURE II. 200 LINES / INCH COMPARISON AND ANALYSIS OF DOT IMAGE SCANNING BEFORE AND AFTER

Figure 1 and Figure 2 indicate that the image dot region has the following characteristics:

1. Although the dot area in the manuscript is clearly defined in black and white, after processing by the scanning optical imaging system of the imaging equipment, the black and white dots are filled with the data in the intermediate order, and the dot shape is changed significantly.

2. The distribution of dots in the manuscript is uniform and periodic. After scanning, the image data distribution also changed regularly, and there is a certain period, as shown in table 2.

In fact, the text and line areas of the manuscript after scanning the output data also has the above changes, i.e. black and white boundary blur, line edge transition is obvious. As shown in Figure 3, the left image is the original, and the image on the right is the output document after scanning.

From the above analysis, we can see that although the dot area and the text and the line area have some degree of blur after the scan processing, there are still significant differences. The data distribution of dot after scanning is distributed in neighborhood pixels, and the text and lines are blurred only at the edge, so they can be effectively segmented by setting different FIR filter detection methods.

TABLE II. CHANGES OF DOT DISTRIBUTION PERIOD BEFORE AND AFTER MANUSCRIPT SCANNING

Original Period	**Scan Resolution**	**Scan data change period**
133 line/inch	300 dpi	3*3 pixels*pixels
200 line/inch	300 dpi	7*7 pixels*pixels

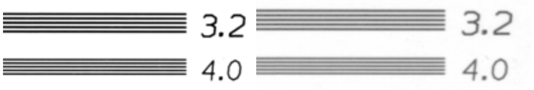

FIGURE III. TEXT AND LINE AREA BEFORE AND AFTER SCANNING CONTRAST

III. THE ALGORITHM OF TEXT-IMAGE SEPARATION BASED ON FIR FILTER

FIR filter is also called Finite Impulse Response filter. The FIR system has permanent stability. FIR filter can not only satisfy the requirement of arbitrary set frequency response, but also obtain strict linear phase characteristic by specifying its frequency response. It is widely used in the field of image processing because of its small distortion in signal processing, the following FIR filter formulas are commonly used:

$$\begin{aligned} y(n) &= \sum_{k=0}^{N} h(k)x(n-k) \\ &= h(0)x(n) + h(1)x(n-1) + \cdots + h(N)x(n-N) \end{aligned} \tag{1}$$

Wherein: *y* is the filter output, *h* is the filter parameters, *x* is the input data. Equation (1) shows that the design of the filter parameter determines the final output effect when the input data is the same.

According to Figure 1 ~ 3 analysis, the dot area after scanning is distributed regularly, and its gradient value is the largest in horizontal and vertical direction. The edge gradient direction of the text and line is scattered. This means that the boundary gradient eigenvalues of the dot region and the text and line regions are significantly compared with the traditional image regions. The difference between the two can therefore be detected by different angles of the gradient. In this paper, we will focus on the design of two kinds of filtering parameters, gradient detection and region conversion.

A. *Gradient Detection*

Gradient detection is based on the difference between the pixel values in the neighborhood. The effect of the detection depends on the difference of the pixel values in the neighborhood and the detection template. According to the characteristic analysis of the data after scanning, the algorithm uses the Prewitt operator to detect the horizontal and vertical direction gradients. The calculation formula is as follows:

$$Edge_0 = \sum_{di=1}^{di=N}\sum_{dj=1}^{dj=N} prewitt_0(di,dj) \cdot src(i+di-\frac{M+1}{2}, j+dj-\frac{N+1}{2}) \tag{2}$$

$$Edge_{90} = \sum_{di=1}^{di=N}\sum_{dj=1}^{dj=N} prewitt_{90}(di,dj) \cdot src(i+di-\frac{M+1}{2}, j+dj-\frac{N+1}{2}) \tag{3}$$

Wherein: *i*, *j* is the location coordinate of the current convolution neighborhood center point. $Edge_0$ and $Edge_{90}$ respectively represent the edge detection intensity values of the image data in both horizontal and vertical directions. The *di*, *dj* is the location coordinate of the current convolution neighborhood point. *M*, *N* is the length and width of the neighborhood. For example, the 3*3 neighborhood, then *M* = 3, *N* = 3. *Src* represents the original image data. The $prewitt_0$ and $prewitt_{90}$ represent a horizontal and vertical edge detection FIR filter, respectively.

The edge strength values of the text line area and image area in the original image are in different regions after edge detection. If the text line area may be between [128, 255], the image area may be between [0, 64]. Therefore, it needs to be effectively separated by setting a simple threshold value. After the segmentation of the text line area strength value is 1, the intensity value of the image area is 0.

$$EdgeStr(i,j) = \left|Edge_0(i,j)\right| + \left|Edge_{90}(i,j)\right| \qquad (4)$$

$$DectStr(i,j) = \begin{cases} 1, EdgeStr(i,j) \geq StrThreshold \\ 0, EdgeStr(i,j) < StrThreshold \end{cases} \qquad (5)$$

Wherein: *EdgeStr* represents the final edge strength value of the image data, *StrThreshold* represents the detection threshold, and *DectStr* indicates the strength value of the output image after the threshold is separated.

In order to verify the effect of gradient detection processing, the text lines, 133lpi and 200lpi dots are processed respectively. The following four images are processed renderings, from left to right respectively: original, horizontal edge detection intensity output image, vertical edge detection intensity output image, threshold segmentation intensity output image.

In the above processing, the edge distribution of the text and line area is basically 360^0 directions. Therefore, the detection in the horizontal and vertical two direction will not only lose the edge feature, but also make it more obvious. Conversely, for the dot region, the result of the final detection is to reduce the edge strength because the data distribution has a fixed law. After the two directions are superimposed on each other, the edges are all lost, i.e. the edge strength is 0. Based on this obvious feature, it can be separated by a simple threshold segmentation method.

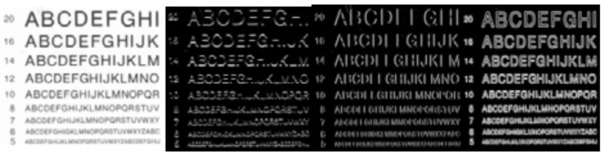

FIGURE IV. TEXT PROCESSING SCHEMATIC

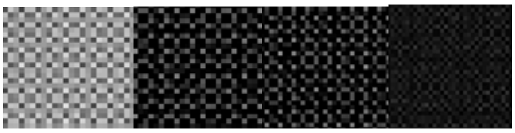

FIGURE V. 133 LPI DOT PROCESSING SCHEMATIC

FIGURE VI. 200 LPI DOT PROCESSING SCHEMATIC

B. *Zone Conversions*

After the gradient detection of the text line area detection results are shown in the Edge form. As shown in Fig.7 on the left, the text and lines appear hollow, not the actual content area. Therefore, it is necessary to convert the boundary region into a content region through the zone conversion processing.

Traditional boundary content conversion requires complex morphological processing (such as corrosion, expansion, etc.)[7], or by judging the boundary direction of the current neighborhood. This type of processing involves multiple convolution, long processing time, and the need for large cache storage of intermediate data. In order to overcome these shortcomings, this paper uses a neighborhood weighted operation, the formula is as follows.

$$Region(i,j) = \left(\sum_{di=1}^{di=M}\sum_{dj=1}^{dj=N} DectStr\left(i+di-\frac{M+1}{2}, j+dj-\frac{N+1}{2}\right)\cdot 255\right) - (M\cdot N-1)\cdot Src(i,j) \qquad (6)$$

$$RegionIndex(i,j) = \begin{cases} 1, Region(i,j) \geq RegionThreshold \\ 0, Region(i,j) < RegionThreshold \end{cases} \qquad (7)$$

Where *Region(i, j)* represents the image data after the fill, and *RegionThreshold* represents the region segmentation threshold. 1 represents the text data after the fill is processed, and 0 represents the non-text area after the fill. Figure 7 is a comparison of the area before and after the conversion of the above algorithm.

C. *Algorithm Description*

Combing with the two algorithms of gradient detection and zone conversion, we can obtain the algorithm of the text-graph separation based on FIR filtering, called TISBFIR algorithm. Its detailed steps are shown in table 3.

In the TISBFIR algorithm, the main parameters to determine the results of text-image separation are as follows:

1. FIR Filter Neighborhood Size: determined by the scanning image resolution, for 300 dpi input of the original data, the use of 3*3 neighborhood can meet the requirements. For 600 dpi, 5*5 can be satisfied.

2. Gradient detection threshold: This threshold determines the segmentation effect of dot area and text and line area. The

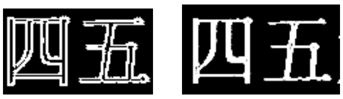

FIGURE VII. COMPARISON CHART BEFORE AND AFTER ZONE CONVERSION

TABLE III. THE TISBIFR ALGORITHM STEPS

Step 1:	Get image data. If the input image is color, you need to turn its data into grayscale data.
Step 2:	Set the detection filter, respectively using the formula (2) and (3) for filtering processing, and the corresponding filtering data is obtained.
Step 3:	According to the formula (4) and (5), the final test result data is obtained. If the result data is 1, the text line area is represented. If 0, it is represented as an image dot area.
Step 4:	Based on the formula (6), the area of the text line detected by the third step is processed by zone conversion.
Step 5:	Set the zone conversion threshold, remove extraneous edge information, and get the final separated text line area and image area.

lower the threshold value, the worse the segmentation effect. Conversely, the more serious the loss of the text line area. Therefore, for ordinary printing and printed manuscripts, the value is generally between [16,80].

3. Zone conversion Threshold: This threshold determines the accuracy of the final text-image separation, and generally takes a value between [32,96].

IV. ALGORITHM EVALUATION

In order to verify the effect of the algorithm, the following is a pair of 300 dpi image mixed grayscale images as input. Set the vertical direction gradient detection filter parameter is [1,0,-1;0,0,0;1,0,-1], the horizontal direction gradient detection filter parameter is [1,0,1;0,0,0;-1,0,-1], the gradient detection threshold is 64, the zone conversion threshold is 80. The processing effect as shown in figure 8. The leftmost figure is the original. The middle shape is the image area of the original after segmentation and dot area. The rightmost graphic is the text and line area after the original is split.

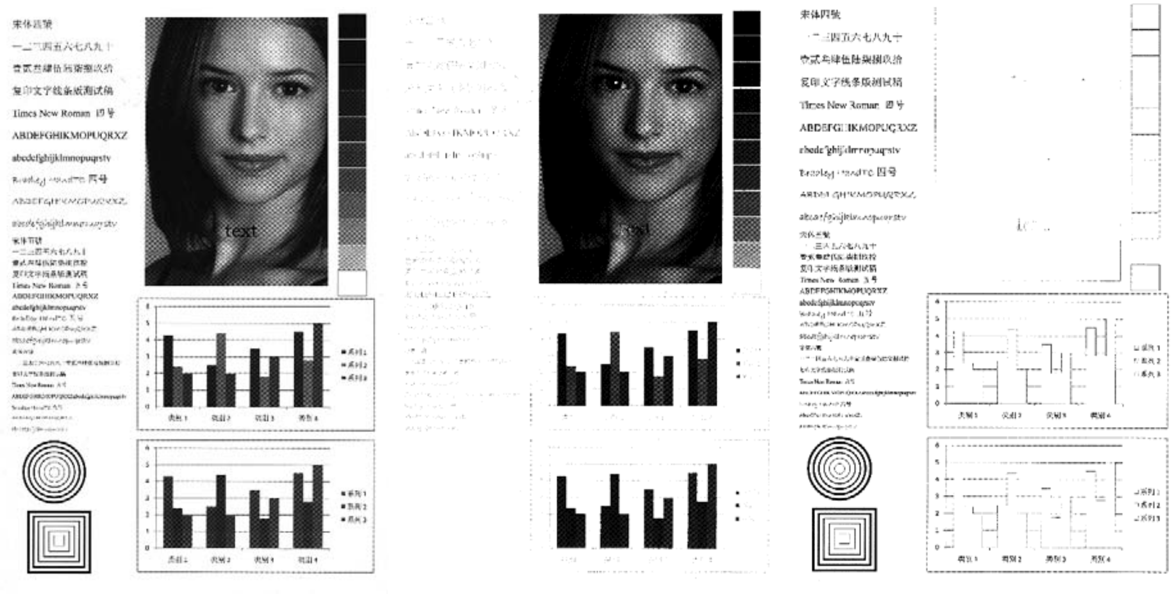

FIGURE VIII. THE PROCESSING RESULTS OF THE MIXED GRAPH

Figure 8 shows that the algorithm has good text-image separation effect. The entire image processing time is approximately 2.1 seconds. Due to the use of only two 3*3 convolution, the time is about 0.6 times the multi-convolution algorithm, can meet the current mainstream copying equipment application needs.

In order to better illustrate the effect of the text-image separation algorithm, this paper uses the recognition rate and processing time two indicators for comparative analysis. The recognition rate is defined as the standard variance of the effect graph data and the original image data after the segmentation recognition process[9~10]. Table 4 lists the results of the multi-convolution (MC) algorithm and TISBIFR algorithm for the separation of text-image from the same manuscripts.

The results from table 4 can be drawn:

1. The recognition rate of TISBIFR algorithm is better than that of MC algorithm for dot sample and image sample.

2. For text sample and image sample, this algorithm is slightly worse than multi-convolution algorithm. However,

TABLE IV. THE RESULTS OF MULTI-CONVOLUTION ALGORITHM AND TISBIFR ALGORITHM

Performance	Algorithm Name	Sample Type			
		Text	Dot	Image	Standard[a]
Text recognition rate	MC	0.7811	-	-	-
	TISBIFR	0.8667	-	-	-

TABLE IV CONTINUE

Image recognition rate	**MC**	-	65.9241	8.7892	-
	TISBIFR	-	4.1873	2.133	-
Processing time (s)	**MC**	3.192	3.89	1.581	12.518
	TISBIFR	1.72	1.798	1.132	6.985

a. Standard speed evaluation sample.

because the text itself is darker, and the internal pixel values are evenly distributed, some differences can be compensated in the subsequent filtering process. Also affected by the dispersion of the toner, the user cannot recognize these minor differences in the text on the final print media, so these differences can be ignored.

3. In the processing speed, whether it is standard speed evaluation sample, or other type of sample, the algorithm in this paper is obviously better than the multi-convolution algorithm.

V. SUMMARY

This paper analyzes the differences between the data features of dot region and text line area page in common hard copy manuscripts, and designs a text-image separation algorithm suitable for gray data based on the FIR filtering principle. The algorithm realizes the complicated graphic and text separation process through the gradient detection and the region transformation, and reduces the hardware resource requirement greatly. In addition, the utility of the algorithm can be further expanded by configuring different detection coefficients to adjust the final processing effect accurately. Finally, the actual verification results show that the algorithm can meet the needs of practical application.

REFERENCES

[1] Weilin ZHAO, Zhenhong JIA, “Grabcut color image segmentation algorithm combined with Bayesian classification and SLIC,” Laser Journal, vol. 38, pp. 84-88, May 2017.

[2] Qiao ZHONG, “Segmentation and correction of scanned image text line based on graph theory,” Hu Nan University Master Thesis. pp. 1-68., April 2017.

[3] Leyuan LIU, Yi ZHAO, Jingying CHEN. “Book page retrieval method based on convolutional neural network,” J. Huazhong Univ. of Sci. & Tech.(Natural Science Edition), vol. 45, pp. 22-28, November 2017.

[4] Qi WANG, Xiaolin QIU, Qian WANG, “Study and appraise on the extraction method of whole tone FM dot microscopic Structure,” Science Technology and Engineering, vol. 15, pp. 37-42, July 2015.

[5] Hu CHEN, Chaodong LING, “Real-time FPGA-based implementation of color image edge detection algorithm,” Chinese Journal of Liquid crystals and displays, vol. 30, pp. 143-150, February 2015.

[6] Renjin LIU, Y B GAO, X G HAO, “An algorithm for document images segmentation,” Journal of university of science and technology of China, vol. 40, pp. 500-504, May 2010.

[7] Ying ZHAO, Z S LIU, S Y LI, “The design of method for digital filter based on Matlab,” Foreign electronic measurement technology, vol. 31, pp. 35-37, October 2012.

[8] Dongxing LI, Q Q GAO, Qi ZHANG, “Edge detection algorithm fused with mathematical morphology technology,” Journal of Shandong University of Technology (Natural Science Edition), vol. 32, pp. 1-5, September 2018.

[9] Jiang CHU, Qiang CHEN, Xichen YANG, “Review on full reference image quality assessment algorithms,” Application Research of Computers. vol. 31, pp. 13-22, Jan.2014.

[10] Lina CHEN, Zhen LIU, “Study of Quality assessment based on halftone characteristics,” Packaging Engineering, vol. 33, pp. 98-101, September 2012.