Detection of Text Regions From Digital Engineering Drawings

Zhaoyang Lu

Abstract—An algorithm for text/graphics separation is presented in this paper. The basic principle of the algorithm is to erase nontext regions from mixed text and graphics engineering drawings, rather than extract text regions directly. This algorithm can be used to extract both Chinese and Western characters, dimensions, and symbols and has few limitations on the kind of engineering drawings and noise level. It is robust to text–graphics touching, text fonts, and written orientations.

Index Terms—Document image analysis, engineering drawings, text segmentation, pattern recognition, image processing.

1 INTRODUCTION

IN recent years, great progress has been made in optical character reader (OCR) technology with which the image to be recognized may be mixed by text strings graphics and gray pictures. In the automatic engineering drawing scan-input system or the document recognition system, the original binary image usually contains more than one kind of information symbol. Because of the great differences in characteristic between these symbols, they are processed in different ways. The understanding of an intensive image, the vectorization of graphics, and the recognition of English letters, digits, and Chinese characters are treated by obviously different strategies. For these reasons, it has become desirable for a computer to separate the text from images.

In this paper the text/graphics separation problem in the recognition of binary engineering drawings has been discussed. By "engineering drawings," we mean mechanical drawings, geographic maps, meteorological maps, electric circuit maps, program charts, etc. Conventional research work on text/graphics separation is described in the literature [1], [2], [3], [4], [5], [6], [7]. In [1], a concept that the connected region of the text is often smaller than that of graphics is used to separate the text from graphics after the thinning process. The method is quite simple and direct, but the method is invalid when a character touches another character or a character touches graphics. Yamada et al. [2] introduce a feature extraction algorithm for topographical maps that can be used to extract routes and buildings from the map, with the rest of the image being considered as text. The method described in [3] is for the separation of the text from the image, including graphics and intensive pictures, but fails when graphics enclose the text. The algorithm reported in [4] may be the best work in text/graphics separation. It is based on the analysis of local connected components and the applications of Hough transform to group components into logical character strings that may then be separated from graphics. It is robust to changes in text font style, size, and orientation. The recent work in [5] discusses the detection of dimension sets that is based on a rule-based algorithm, with the detection of arrowheads also considered.

In summary, most of the methods mentioned above are based on one or two of the assumptions listed below:

• *The author is with Group 102, Institute of Telecommunications Engineering, Xidian University, Xi'an, 710071, People's Republic of China. E-mail: zhylu@xidian.edu.cn.*

Manuscript received 29 Aug. 1995; revised 13 Mar. 1998. Recommended for acceptance by R.. Kasturi.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106569.

1) The original digitized images should not have many noise components, and they should be prepared according to some standard as precisely as possible.
2) The text should be printed or hand-printed, and it should seldom touch graphics. It is preferably written vertically or horizontally, rather than in any other direction.
3) The text should not contain any Chinese characters.

These assumptions limit the applications a great deal. For example, in CAD/CAM applications, it is necessary to automatically convert plenty of existing paper drawings into a standard CAD/CAM format. Those paper drawings are occasionally blue-printed and suffer from many noise components. The character-with-character and character-with-graphics touches are also unavoidable. The interactive modification will have to be made, which is labor-intensive and time-consuming.

In this paper, a new rule-based method for text/graphics separation is presented based on the analysis of text and graphics features of general engineering drawings. The main strategy of the proposed method is to erase nontext regions from engineering drawings as much as possible, rather than extract text regions directly. This algorithm can be used to extract both Chinese and Western characters, dimensions, and other special symbols from a text/graphics mixed image quite successfully. It has few limitations on the kind of engineering drawings and noise level and is also robust to text–graphics touching and text fonts. The written orientations of text strings can also be evaluated. Since thinning processing will certainly cause image information loss, the algorithm is performed at the pixel level before thinning and vectorizing processes, so that two images can be obtained—one containing text strings and the other graphics—which can then be processed by suitable character and graphics recognition systems.

In the following, the details of the proposed algorithm are given. In Section 2, we summarize the general features of engineering drawings. Section 3 describes how to delete graphics components step by step. Sections 4 and 5 give the experimental results. Section 6 concludes this paper.

2 General Features of Engineering Drawings

What is meant by "text" or "graphics" actually? It seems that there is no objective discrimination between them for a digitized binary image, since both of them are mathematically "0" or "1" sequences distributed on a 2D plane. Thus the distinction between text and graphics might be in a sense subjective and statistical. It is particularly true when the text is composed of Chinese characters. Some examples of definition confusion are shown in Fig. 1, where region 1 is similar to region a, 2 to b, and 3 to c. Although it is rather difficult for a computer to differentiate some text from graphics like Fig. 1, it is very easy for us to classify them as the following:

- *Text*—symbols or strings for interpretation or illustration, including letters, words, digits, Chinese (or other language) characters, and/or special symbols.
- *Graphics*—nontext components left in engineering drawings, including all kinds of lines, curves, and/or pictures.

On the basis of observations of typical engineering drawings, we have found that the geometric features of text regions and graphics differ in aspects listed below:

1) The size of text characters is often much smaller than that of graphics. Changes in text size or shape are within a narrow range. As for mechanical engineering drawings prepared according to the Chinese GB-126 drafting standard, the Chinese characters on the drawings have the width/height ratio of about 2:3.
2) Text characters usually appear in the form of strings. Gaps between characters or words are relatively small and have regular patterns. The orientations of strings are often horizontal, vertical, or slanted at an angle of 45 degrees.
3) The local stroke density of text regions is often much higher than that of graphics, so is the time of alternate changes of local strokes.
4) The length of the linear components included in strokes of text strings is much shorter than that of graphics, i.e., long straight lines rarely appear in the strokes of text strings.

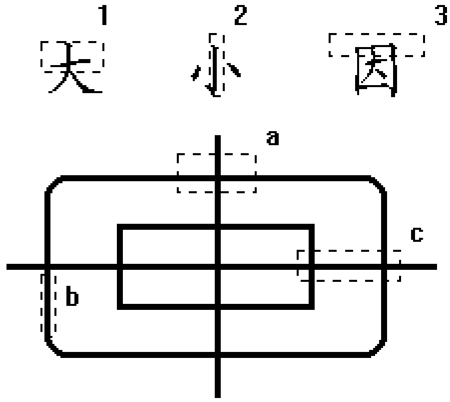
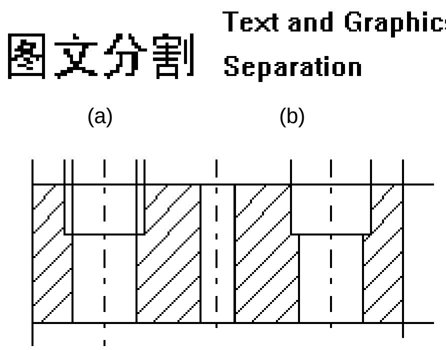


Fig. 1. Definition confusion of text and graphics.



(a) (b)

(c)

Fig. 2. Example of text and graphics.

TABLE 1
Local Stroke Density (LSD) of Fig. 2

	Width	Height	Black pixels	White pixels	B/(B+W) (%)
"图"	12	15	88	92	48.9
"文"	14	15	59	151	28.1
"分"	15	15	62	163	27.6
"割"	13	15	84	111	43.1
"Text"	28	10	98	182	35.0
"and"	22	10	94	126	42.7
"Graphics"	56	13	220	706	30.2
"Separation"	68	13	268	616	30.3
Fig. 2. (c)	201	92	1801	16691	9.7

Naturally, each feature described above has its exceptions. However, those features reflect general situations.

As an example, two kinds of text, Chinese characters and English words, and a typical graph are shown in Fig. 2. "图文分割" means "Text and Graphics Separation." The local stroke density (LSD) of the text and the graph is shown in Table 1. Let B be the number of black pixels counted in an observation region and W be that of white pixels, and then LSD is defined by B/(B + W) in percentage. The observation regions in Table 1 are the circumscribing

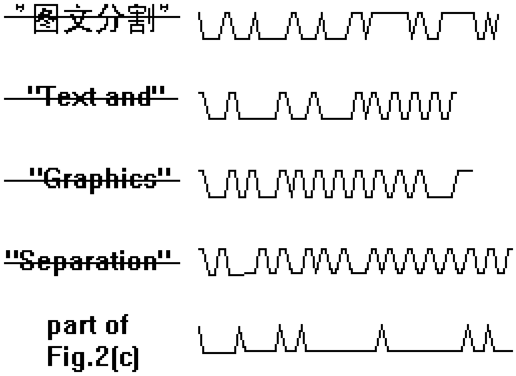
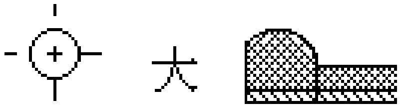


Fig. 3. Intersecting pixel series (higher level represents black pixels).



(a) (b) (c)

Fig. 4. Typical exceptions.

rectangles, with a rectangle containing a Chinese character, a rectangle containing an English word, and a rectangle containing the whole graph of Fig. 2c, respectively. If we draw a horizontal line crossing each character string and graphic of Fig. 2, the intersection between these lines and strokes will be a black and white pixel series. Suppose that the higher level represents black, and the lower level represents white. The intersecting pixel series is shown in Fig. 3. As shown in Fig. 3, the alternate change's frequency of the strokes of the text is often higher than that of graphics. Fig. 4 shows some typical exceptions. Fig. 4a is an isolated graph of small size which is similar to the isolated text. Fig. 4b is an isolated Chinese character "大" ("big") with few strokes, which will be hard to distinguish from graphics, especially when it touches them. Fig. 4c is a graphics region with a higher stroke density. These kinds of exceptions are the main reason for the error separation in the experimental results as discussed below.

3 Algorithm of Text/Graphics Separation

A rule-based algorithm of text/graphics separation using "deleting graphics and making the text left" strategy is described in this section. A block diagram of the algorithm is shown in Fig. 5. The main steps of the algorithm are recapitulated below.

3.1 Erasion of Linear Components

According to feature 4 of Section 2, the linear components of the original image could be erased first. By "linear component (LC)," we mean the consecutive black pixels when tracing along specific directions. In this procedure, a top-to-bottom row-by-row scanning is performed to find the horizontal LCs, which will be deleted if their lengths are longer than a preset threshold T1. Then the same will be done to vertical LCs by a left-to-right scanning. LCs with ±22.5, ±45, and ±67.5 degrees will also be removed.

Suppose the narrower edge of the circumscribing rectangle of any text has a thickness of more than one pixel, and then it is appropriate to discard isolated short horizontal or vertical straight lines of one pixel thickness. So most noise points (or short LCs) will be removed in this step.

The method for detecting straight lines other than vertical or horizontal directions is illustrated in Fig. 6. A simple geometric operation of stretching is performed first. We calculate an intercept

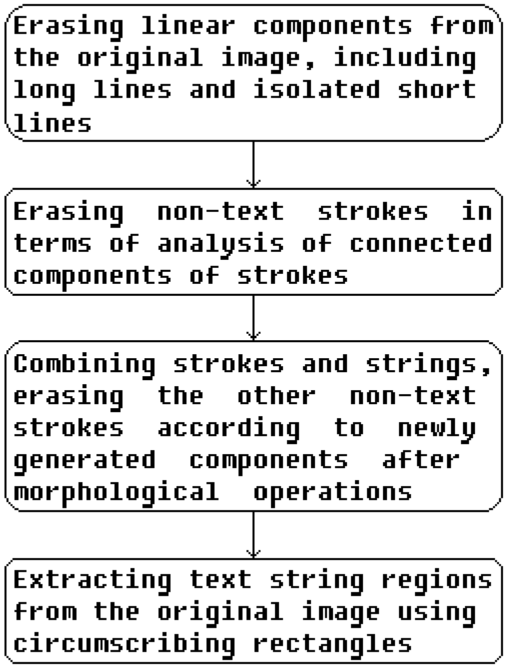
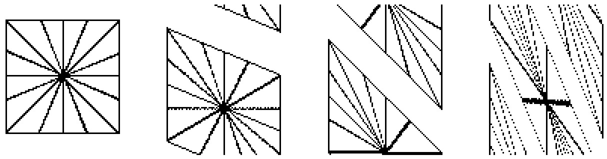


Fig. 5. Block diagram of text and graphics separation.



(a) (b) (c) (d)

Fig. 6. Stretching operation for slant line detection. (a) Original image. (b) α = +22.5. (c) α = +45. (d) α = +67.5.

value B for every column j by B = (tgα×j)%H, where j = 1 ~ W and α is the angle of slant lines to be detected. H and W are the Height and Width of the image, respectively. Then all the pixels in this column j are moved down for B pixels. That is, the pixels (i, j) are moved to (B + i, j) for i = 1 ~ H-B, and (i, j) to (1, B) for i = H-B ~ H. Here (i, j) represents the line and column position, $1 \leq i \leq H$ and $1 \leq j \leq W$. Fig. 6b is the stretching result of Fig. 6a when α = 22.5 degrees. Fig. 6c is for α = 45, and Fig. 6d for α = 67.5. The other three cases for α = –22.5, –45, and –67.5 degrees are not listed. Obviously, the slant lines having 22.5 degrees in Fig. 6a will remain horizontal in Fig. 6b. Then we can do horizontal scanning to find and erase these lines. The threshold for deciding whether the line is long enough or not should be adjusted to T1 × cosα, because the stretching operation changes the length of the lines. After all that, an exactly opposite stretching operation is performed to recover the original image. A pair of stretching transforms does not modify anything except the erased lines, we expected.

If the paper drawings are carefully placed when digitized, most of the horizontal or vertical lines in the original image will stay at strictly zero or 90 degrees, and slant lines will stay at 45 degrees. So enormous amounts of black pixels that belong to graphics will be removed in the above procedure, which is very helpful for speeding up the following steps. For a small number of slant lines other than zero, ±22.5, ±45, ±67.5, or ±90 degrees, probably existing in the image, it is preferable to leave them to the next processing rather than to delete them in order to save the CPU time.

3.2 Analysis of Connected Components of Strokes

There are still graphical components, which are different in size, left after the erasion of linear components. In this procedure 3.2, we process each connected component encountered in scanning the whole image row by row. "Connected component" [4] (CC) means black pixels belonging to text regions or graphics which are eight connected to one another. Although the method of treating CCs in this paper is similar to that of [4], they are different in the following aspects.

- First, the CCs we process are generated in the image after the erasion of LCs other than the original image as in [4].
- Second, we use stroke density information of CCs to identify graphics.

Once the CC is accepted as graphics, it will be deleted immediately. We do not have to restore the information array of CCs as in [4].

In this procedure, the coordinates' sequence of boundary points of each connected component is generated by edge tracing, and it is expressed as (Xi, Yi), i = 1, 2, ..., n, where n is the total number of boundary points. The following important parameters are then calculated:

1) MaxBox, which represents the maximum and minimum coordinates of the circumscribing rectangle of the CC;
2) WBRatio, which represents the stroke (black) pixel density in the circumscribing rectangle MaxBox; and
3) HWRatio, which represents the dimensional ratio of the circumscribing rectangle MaxBox. If HWRatio < 1, set HWRatio = 1/HWRatio.

CC is considered as graphics if the above parameters satisfy one of the following conditions:

A) WBRatio $\leq$ T2,
B) HWRatio $\geq$ T3, or
C) the length of the longer edge of MaxBox $\leq$ T4.

Here, T2, T3, and T4 are preset thresholds like T1. Condition A is used to delete slant lines other than zero, 22.5, 45, 67.5, or 90 degrees from the image, since the number of background (white) pixels in the MaxBox of any slant line CC is certainly much larger than that of text CC. Using Condition B, the graphical components with long and narrow shapes could be removed. By Condition C, we can also erase the graphical component that is obviously smaller than a text character, such as image noise.

After this procedure, more graphical components disappear. Some text information may be deleted as well, such as decimal points and isolated strokes of some Chinese characters. However, a decimal point, which is very similar to noise, could be recovered by subsequent procedures if there is a context before and after it. The loss of strokes of Chinese characters may also be compensated for in the following steps.

3.3 Combination of Strokes and Strings

In terms of feature 4 of Section 2, the length of a gap between character strokes or character strings is usually small, so that they could be grouped together by combination. If two black pixels are near enough, i.e., the number of white pixels between them is smaller than a preset value T5, we will fill the gap between them with black pixels. It is just like "brushing" them with a black pen. The directions of brushing are horizontal and vertical. Fig. 7 is an example of brushing operations. The procedure aims at combining strokes or characters into new connected components (NCCs), which is practical due to feature 2 mentioned above. The purposes of this procedure can be summarized as follows:

1) The decimal point, the symbol like "-," and some isolated strokes that are erased in preceding procedures can be recovered.
2) A combination of strokes and characters facilitates text recognition, since the text strings being combined in an NCC are probably context sensitive.
3) The brushing operation above makes graphical components isolated, which enables us to remove them more easily in subsequent procedures.

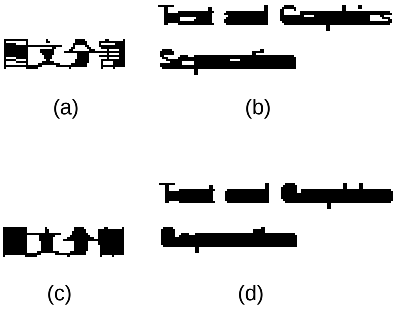


Fig. 7. Brushing operations (T5 = 4). (a) and (b) Horizontal brushing of Fig. 2a and Fig. 2b. (c) and (d) Vertical brushing of (a) and (b).

3.4 Morphological Operation

This procedure consists of one erosion operation and one dilation operation. The purpose of the morphological operation is to eliminate possibly existing character–graphics connections or unnecessary text string touches and rebuild the NCCs generated in procedure 3.3 to facilitate the final separation of text and graphics.

3.5 Analysis of New Connected Components

This procedure is similar to procedure 3.2, though it deals with NCCs rather than CCs. In a similar manner, we trace the boundary of each NCC encountered in scanning the image row by row and generate the coordinates' sequences of boundary points. Three parameters, MaxBox2, WBRatio2, and HWRatio2, are calculated. The NCC can be considered as a graphical component if these parameters satisfy one of the following conditions:

- WBRatio2 $\leq$ T6,
- the length of the longer edge of MaxBox2 $\leq$ T7,
- The length of the shorter edge of MaxBox2 $\leq$ T8, or
- The length of the shorter edge of MaxBox2 $\leq$ T7 and HWRatio2 $\geq$ T9.

Here, T6, T7, T8, and T9 are all preset thresholds.

The written orientations of text strings are estimated in this procedure. The method we use is to try to minimize the circumscribing rectangle of the NCC if its WBRatio2 is not high enough, say, lower than 75 percent. For this reason, the boundary image of the NCC is rotated by a certain number of degrees to find the minimum circumscribing rectangle. The orientations are quantized with every 10 degrees in our algorithm, so that we need eight rotations for each NCC. The minimum rectangle is then reversely rotated by the same number of degrees, with the result that the resulting slant rectangle will be a good estimation of the minimum circumscribing rectangle of the original NCC, and the angle of this rectangle will be the approximate orientation of the text.

After this procedure, all graphical components are deleted as much as possible.

3.6 Extraction of Text Strings

In procedure 3.5, when an NCC is accepted as a text string, its circumscribing rectangle MaxBox2 will be registered into the following structure:

Char_Box { double Angle; int Coe[4]; },

where Angle represents the orientation of the text string. When Angle = 0, the text region is a rectangle, and Coe[4] represents

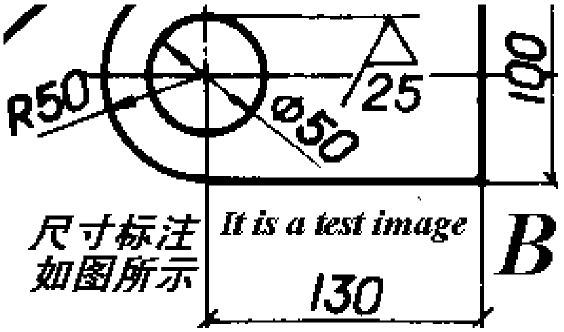


Fig. 8. Original engineering drawing as a test image.

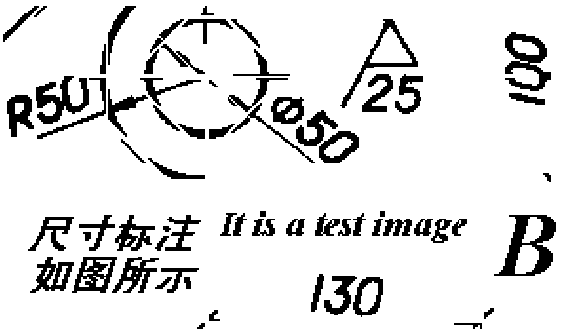


Fig. 9. Output after the erasion of linear components.

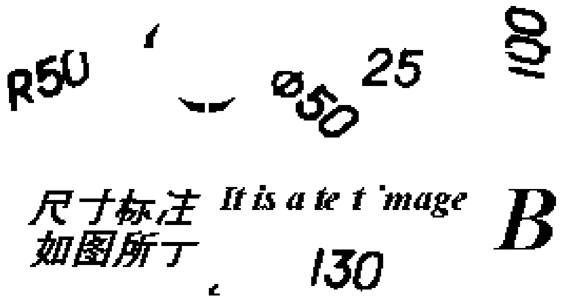


Fig. 10. Output after the analysis of connected components of strokes.

the coordinates of the upper left and lower right corners. When Angle > 0, the text region is a slant rectangle, and Coe[4] represents the coordinates of the left-most and right-most corners. For both cases, Coe[4] is slightly adjusted according to the result before the brushing procedure, i.e., step 3.3, because the NCC was probably modified by the above morphological operation. Consequently, the extraction of text regions means the extraction of the registered rectangular regions from the original image. If we subtract text regions from the original image, the remainder will be graphical parts.

3.7 Parameter Setting

We have described the algorithm for text/graphics separation step by step. It is easy to find out that this rule-based method needs as many as nine preset thresholds T1 ~ T9 to help the classification of text and graphics. How to preset these nine values affects the performance of separation greatly. Each of the parameters could be set alone or adjusted according to certain properties of the original image automatically. Through experiments on typical drawings, better selection could be obtained. The ways we have used in this paper are as follows:

$$T1 = 2 \times H_{av};\ T2 = 0.3;\ T3 = 4.5;\ T4 = 0.5 \times H_{av};\ T5 = 0.6 \times H_{av};$$

$$T6 = 0.3;\ T7 = 0.7 \times H_{av};\ T8 = 0.25 \times H_{av};\ T9 = 5.$$

Fig. 11. Output after the combination of strokes and strings.

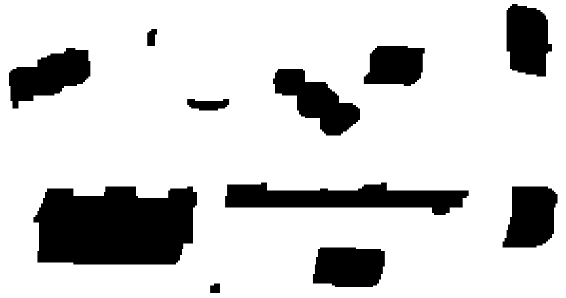


Fig. 12. Output after the morphological operations.

Except constants T2, T3, T6, and T9, all thresholds depend on H_{av}, which is the average height of characters counted in the number of pixels in an image. This means that only one value, H_{av}, is needed to set these parameters. H_{av} can be set manually or estimated automatically by using the histogram method of [5]. In the following experiments, H_{av} has been set manually. The selection of the parameters on the basis of the average height of characters is convenient for applications and is especially effective when the paper-based documents are mechanical engineering drawings. We choose the parameters such that they not only assure good performance of the algorithm but also are of general purpose.

4 Illustrations of Every Process Step

A typical test image of 292 × 170 pixels shown in Fig. 8 is used to illustrate the various steps of the process. In order to evaluate the performance of the algorithm discussed above for different text sizes and for Chinese characters, the letter "B," the sentence "It is a test image," and eight Chinese characters are interactively inserted by using Windows PaintBrush tools. The image shown in Fig. 8 has several characteristics found in different types of mixed text/graphics images:

- different font styles and various sizes (with the height of the character ranging from nine to 32 pixels);
- text strings enclosed by graphics;
- text strings with various orientations;
- text/graphics touches; and
- Chinese characters.

The results of every processing step of our algorithm are shown in Figs. 9 to 14. H_{av} is manually set to 20 pixels, and the parameters T1 ~ T9 are then computed according to Section 3.7.

Fig. 9 illustrates the output after the erasion of linear components, including the LCs of zero, ±22.5, ±45, ±67.5, and ±90 degrees. The isolated short horizontal and vertical lines and most noise points are also removed in this procedure. Some strokes of Chinese characters have been deleted, because they are "isolated straight lines."

Fig. 13. Output after the analysis of new connected components.

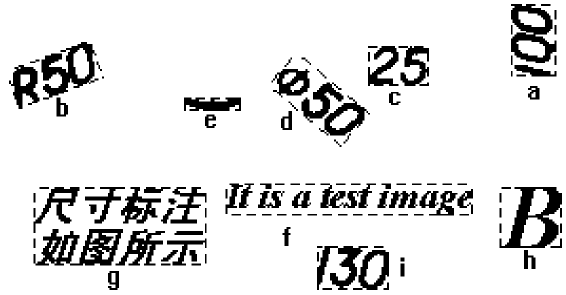


Fig. 14. Final result of the separated text.

TABLE 2
SUMMARIES OF THE EXAMPLES

	Document type	Image size	H_{av}	No. of TEXT regions	TEXT region error	GRAP region error	Process time (P-133) (second)
Fig. 13	Mecanical drawing	292×170	20	9	1	0	3.13
Fig. 15	Mecanical drawing	560×480	12	44	8	5	15.21
Fig. 16	Frame diagram	628×440	12	48	4	1	16.15
Fig. 17	Mecanical drawing	620×672	13	40	13	0	24.00

Fig. 10 shows the output after the analysis of CCs of strokes. More graphical components that are obviously nontext are removed. Of course, some text strokes also disappear.

Fig. 11 shows the output after the combination of strokes and strings. By this step, the removed strokes are combined with their context.

Fig. 12 is the result of the morphological operation. As shown in Fig. 12, the NCCs have a better shape than that in Fig. 11, which makes it easier to decide the text or graphics.

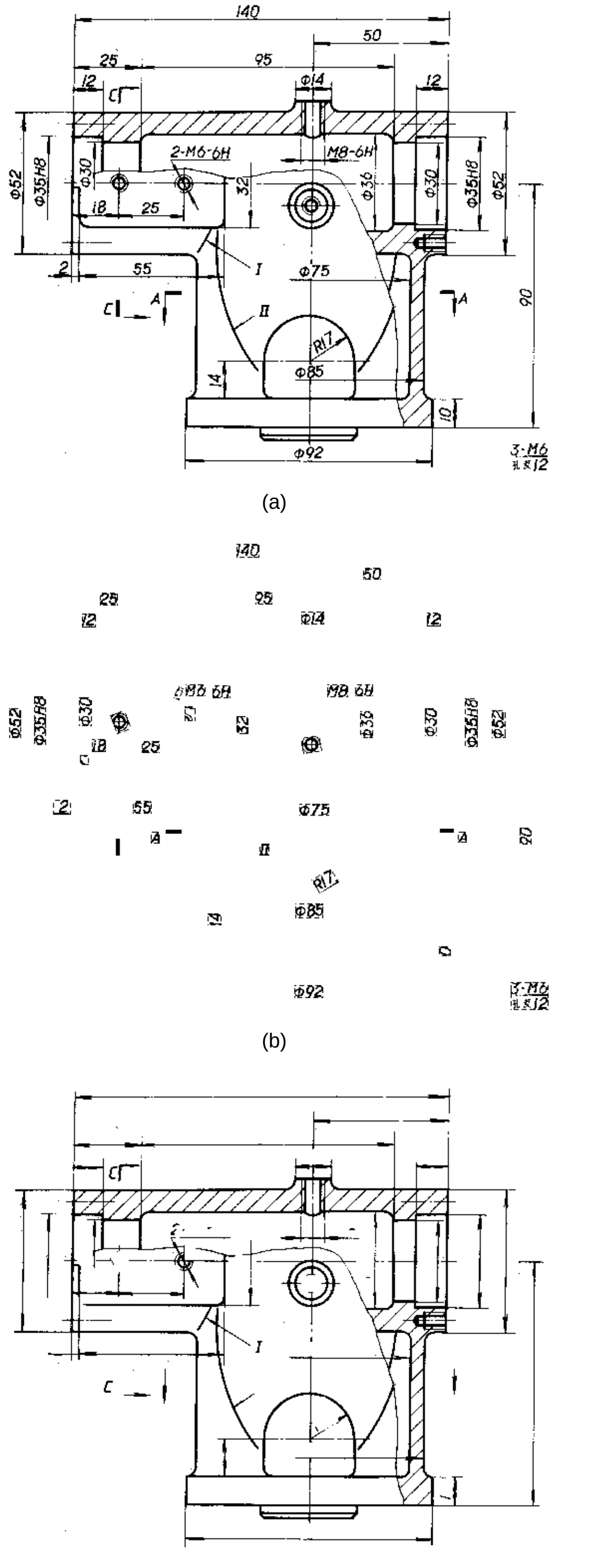
Fig. 13 is the final output after the analysis of NCCs.

Fig. 14 shows the final result of the separated text. We have obtained nine text regions represented by a to i from the test image, in which regions b and d are slanted rectangles of 20 and 50 degrees, respectively. Region e is obviously an error-separated region, because it is a part of the circle in Fig. 8 instead of the text. Since region e has great similarity to the isolated digit "1" or letter "l," we prefer to classify this kind of NCC as a text region.

5 MORE EXPERIMENTAL RESULTS

We have described the algorithm and illustrated the various steps of the process. Here are more examples to demonstrate the utility of the algorithm. Fig. 15, Fig. 16, and Fig. 17 show three results for different document types. We have arranged them in the same manner as the following:

- Part a is the original image;
- Part b is the final result of separated texts; and
- Part c is graphics left.



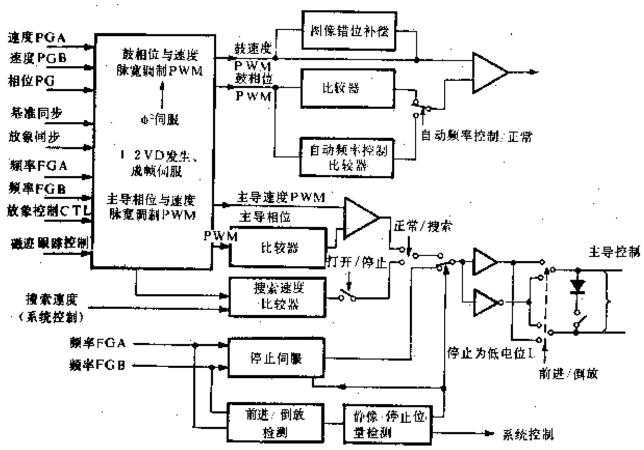
(a)

(b)

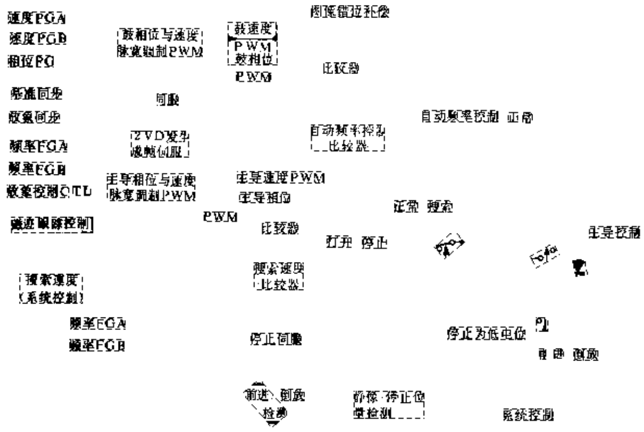
(c)

Fig. 15. Example of a mechanical drawing. (a) Original image. (b) Text separated from the drawing. (c) Graphics left.

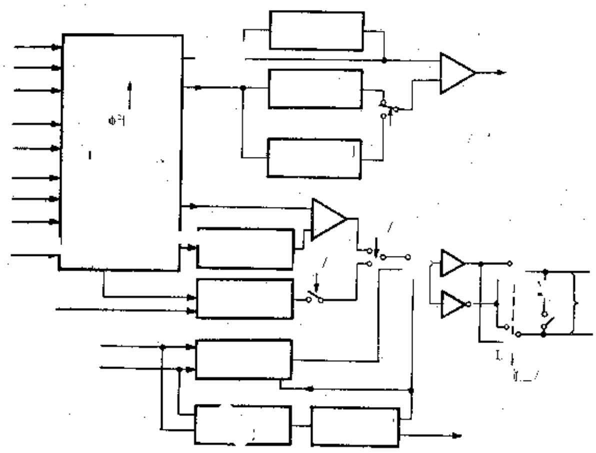
Table 2 summarizes the above results, with the illustration in Section 4 also listed.



(a)



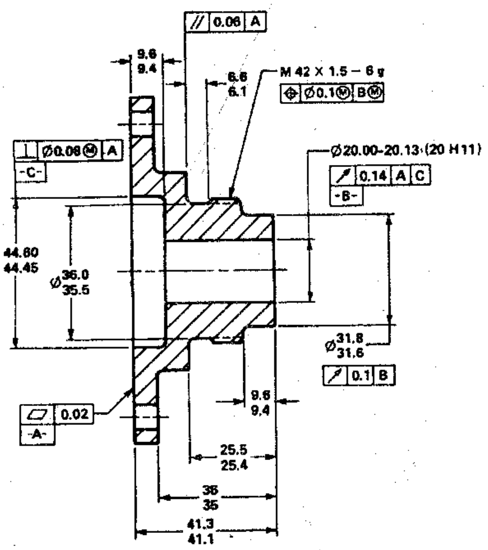
(b)



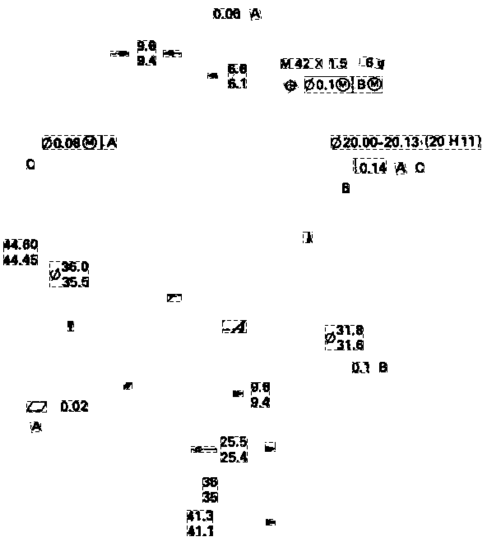
(c)

Fig. 16. Example of a mechanical drawing. (a) Original image. (b) Text separated from the drawing. (c) Graphics left.

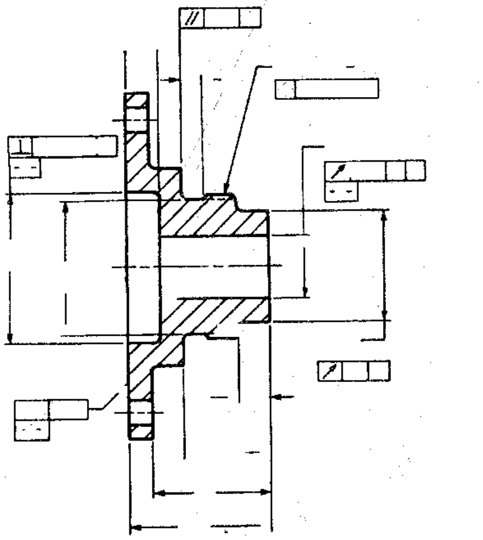
Fig. 18 shows the text segmentation results of Fig. 15a when quantity H_{av} is set to six, 10, 15, and 18, respectively. As shown in these figures, H_{av} could change within a reasonably lage range. We prefer to specify it manually by rough estimation, even though the automatic histogram analysis method [5] could possibly yield a more reliable value.



(a)

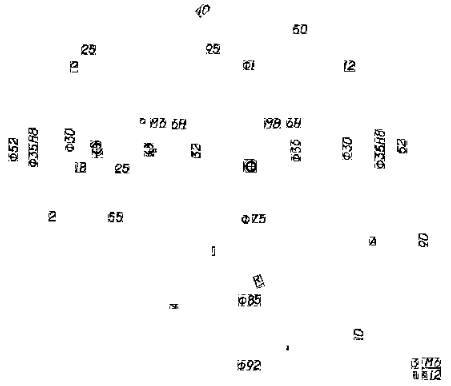


(b)

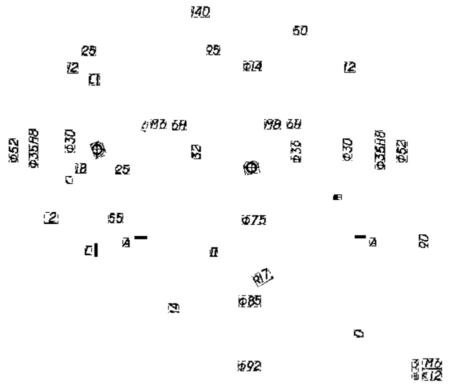


(c)

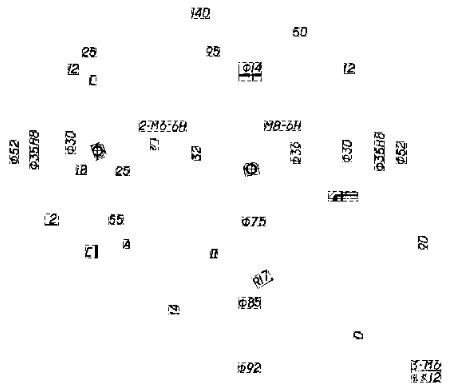
Fig. 17. Example of a frame diagram. (a) Original image. (b) Text separated from the drawing. (c) Graphics left.



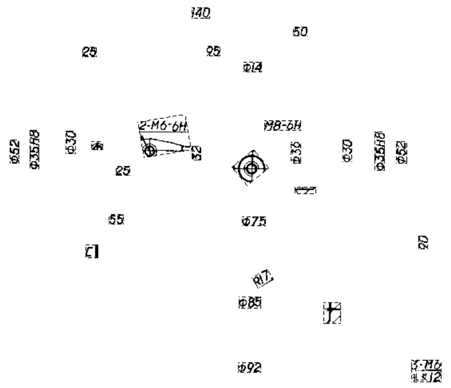
(a)



(b)

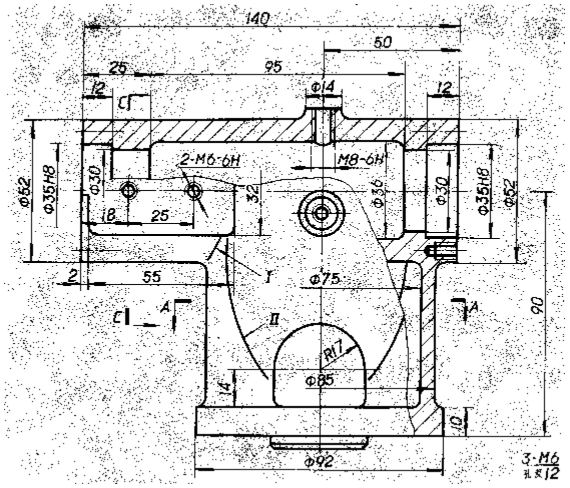


(c)

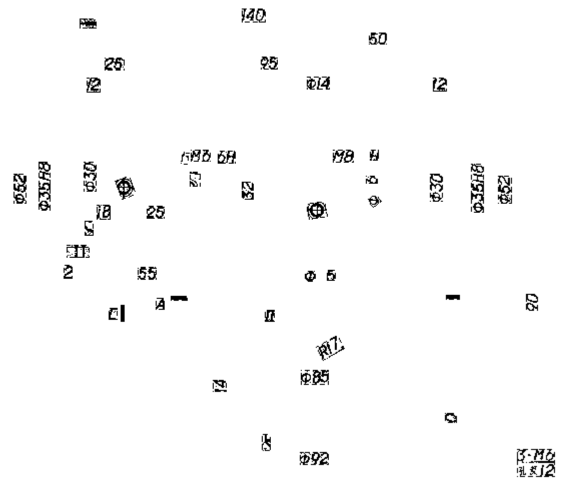


(d)

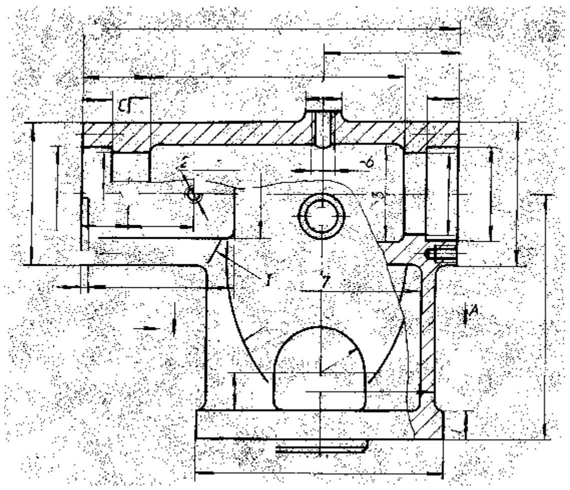
Fig. 18. Text segmentation results of Fig. 15a when H_{av} changes. (a) H_{av} = 6. (b) H_{av} = 10. (c) H_{av} = 15. (d) H_{av} = 18.



(a)



(b)



(c)

Fig. 19. Text segmentation results of Fig. 15a when noise is added. (a) Original image. (b) Text separated from the drawing. (c) Graphics left.

Fig. 19 shows the noise adaptability of the algorithm, where H_{av} remains the same as Fig. 15. The original image of Fig. 15a is blurred by the random distributed noise, and the text-separation result remains almost the same.

6 Discussions and Conclusions

We have described the algorithm and given several examples to illustrate its efficiency and performance when separating engineering drawings of different kinds and sizes. It seems that the algorithm is quite good in the following aspects:

1) It could deal with the special case of text and graphics touching, which is a very hard task for the other text- and graphics-separation algorithm [1].
2) The performance for separating Chinese characters is fairly good.
3) The detection of the writing direction of text strings works well.
4) Most parameters of the algorithm can be calculated according to quantity H_{av}, which is the only thing that needs to be input manually. So the algorithm is rather convenient to use. The performance of the algorithm is not very sensitive to H_{av} if it falls in a limited range around the real average text height.
5) The algorithm is not sensitive when noise is reasonably strong.

It is easily observed that the algorithm does have some breaking points. Table 2 shows two kinds of error: One is "the error-separated text," that means that a graphical component is considered as a text region; the other is "the error-separated graphics" that means that a text region is unable to be extracted from an image. We summarize these two kinds of failure as follows:

1) When the stoke density of graphics is too high or that of text regions is too low, errors may occur.
2) Sometimes a part of a circle may be classified by mistake as a text region, because the linear-component erosion procedure fails to eliminate it.
3) Some isolated text symbols like "•," "1," "l," or "I" or graphics like dash lines are always unreliable when separated because of the definition confusion mentioned in Section 2.
4) If two text strings placed in parallel are excessively closed, they may be classified as a whole text. Because the algorithm uses the rectangular box to extract the text, all components that fall in the box will be separated as a text region. An error will occur if some graphical parts fall in the box.

In summary, an algorithm for text-string separation from mixed text/graphics images has been proposed. Based on the analysis of differences in features of text and graphics, the algorithm accommodates the changes in drawings' types, text font style, size, and orientation within an image. This algorithm can extract both Chinese and Western characters, dimensions, and other special symbols quite successfully. It has few limitations on noise level and is also robust to text–graphics touching. With improvements made to the algorithm in terms of processing efficiency, the application of the algorithm will become highly appropriate for use in office automation systems, OCR, and CAD/CAM applications.

Acknowledgments

The author would like to thank Ying Daoning, Wang Erjian, and Wang Yun of Mold High Technology Institute, Zhejiang University, Hangzhou, China, and Wu Chengke of Xidian University, Xi'an, China, for their support to this project.

References

[1] D.N. Ying, E.J. Wang, L. Ye, W. Li, and Y. Wang, "A Study on Automatic Input and Recognition of Engineering Drawing," *Proc. CAD/GRAPHICS*, pp. 478-481, Hangzhou, China, 23-26 Sept. 1991.

[2] H. Yamada et al., "MAP: Multi-Angled Parallelism for Feature Extraction From Topographical Maps," *Pattern Recognition*, vol. 24, no. 6, pp. 479-488, 1991.

[3] F.M. Wahl et al., "Block Segmentation and Text Extraction in Mixed Text/Image Documents," *CVGIP*, vol. 20, pp. 375-390, 1982.

[4] L.A. Fetcher and R. Kasturi, "A Robust Algorithm for Text String Separation From Mixed Text/Graphics Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910-918, 1988.

[5] C.P. Lai and R. Kasturi, "Detection of Dimension Sets in Engineering Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 848-855, 1994.

[6] D. Dori and Y. Velkovitz, "Separation of Text From Graphics I: Engineering Drawings," *Preproceedings Int'l Workshop Graphics Recognition*, Pennsylvania State Univ., Aug. 1995.

[7] D.B. Lysak and R. Kasturi, "Interpretation of Engineerings of Polyhedral and Non-Polyhedral Objects," *Proc. ICDAR*, 1991.