CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 56, No. 5, September, pp. 402–413, 1994

A Morphological Approach to Text String Extraction from Regular Periodic Overlapping Text/Background Images

SU LIANG AND M. AHMADI

Department of Electrical Engineering, University of Windsor, Windsor, Ontario, Canada N9B 3P4

AND

M. SHRIDHAR

Department of Electrical and Computer Engineering, University of Michigan—Dearborn, Dearborn, Michigan 48128

Received December 8, 1993; revised June 6, 1994; accepted June 13, 1994

A digitized image that consists of text strings and uniformly distributed background symbols must be segmented if the characters in the text string are to be recognized. This paper describes the development and implementation of a morphological approach to character string extraction from overlapping text/background images that minimizes the shape distortion of characters. The effectiveness of this algorithm is demonstrated on several test images. © 1994 Academic Press, Inc.

1. INTRODUCTION

The Optical Character Reader (OCR) is a common tool used to recognize characters. However, most OCR systems can only read traditional characters: black characters on a uniform white background, or vice versa. However, characters are often printed over complex backgrounds. Examples can be found in mail pieces where the address is written on pattern papers [1], the headlines of a Japanese newspaper [2], and headlines of a Chinese newspaper (see Fig. 1). Some text blocks that are decorated with uniformly distributed graphical symbols in their background are designed to help the reader to distinguish various parts of text and to make some articles in a text more attractive. Figure 2 shows an article of *NewsLines* of University of Windsor, which contains text printed on a regular periodic background of dots. Even if some unfamiliar symbols overlap the character strings, humans have no difficulty reading these articles. People can differentiate graphical symbols from text, and instinctively remove the background symbols without difficulty. However, it is not simple for a computer to distinguish text strings from background symbols. The whole text block usually becomes unreadable and is incorrectly recognized as a graphic and removed from the document image by the text recognition system. Therefore, it is necessary to perform a preprocessing procedure to extract the text before a recognition algorithm is applied.

The extraction of text from various kinds of images in which it touches and intersects linework [3], scratch [4], noise background [5], or geometric background patterns [2, 5–7] have become subjects of extensive research. Thresholding [7–9] is a popular tool for segmenting gray level images. The approach is based on the assumption that objects and background pixels in the image can be distinguished by their gray level values. By appropriately choosing a gray level threshold between the dominant values of object and background intensities, the original gray level image can be transferred into a binary form so that the image pixels associated with the objects and the background obtain values 1 and 0, respectively. White and Rohrer [7] described an image thresholding technique based on boundary characteristics to suppress unwanted background patterns so that only printed or handwritten characters may be captured as electronic images. Liu *et al.* [10] proposed a new scheme by using the underlying properties, such as run-length histogram and texture attributes, to correctly binaries the document images. Yamada *et al.* [6] developed a recognition method for characters stamped on metal. They attempted a local binarization after smoothing the uneven background to generate features for distinguishing figures or characters from backgrounds. Billawala *et al.* [4] described a technique called the image continuation algorithm to remove the scratches and blemishes in binarized images. Ozawa *et al.* [2] proposed a method to remove the geometric pattern background in Japanese newspaper headlines.

In this paper, we are especially interested in binarized documents containing text with a regular periodic overlapping background. The proposed algorithms that perform text character extraction accommodate document images

1049-9652/94 $6.00

1994年4月26日 星期二 第7版

人民日报

PEOPLE'S DAILY

OVERSEAS EDITION

1993年12月24日 星期五

癸酉年十一月十二 第2791号

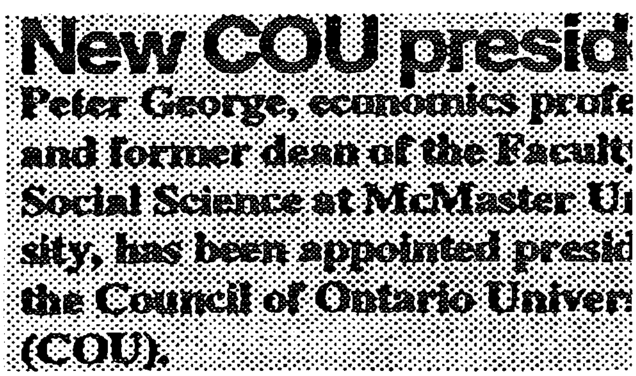
李鹏考察三峡工程施工准备情况时指出

建设好长江三峡工程

关键是人民理解支持

强调运用社会主义市场经济办法引入竞争机制

FIG. 1. Examples of Chinese newspaper headlines.



New COU presid

Peter George, economics profe

and former dean of the Facult

Social Science at McMaster U

sity, has been appointed presid

the Council of Ontario Univer

(COU).

FIG. 2. An image sample with overlapped text/background (from *Newslines* of University of Windsor).

that contain various kinds of background symbols. Mathematical morphology, because of its ability to grasp the geometry and structure of images, is adopted to realize this new scheme. The effectiveness of the proposed algorithms is demonstrated by several experiments that we have conducted.

2. BASIC MORPHOLOGICAL TOOLS

Many theoretical results concerning the mathematical morphological operations can be found in [11–17]. These operations have been applied successfully to a large variety of image processing and analysis applications, such as biomedical image processing, cellular automata, automated industrial visual inspection, and etc. [11]. For instance, basic morphological operations can be used for noise suppression [17], texture analysis and image enhancement [8], and shape analysis [16].

Mathematical morphology is a set-theoretic approach to image processing and analysis that considers images to be sets in underlying space and manipulates them using set-based operations such as union and intersection. The fundamental operations, erosion and dilation, are implemented by "AND"ing or "OR"ing the images that have been translated by structuring elements to generate eroded or dilated images.

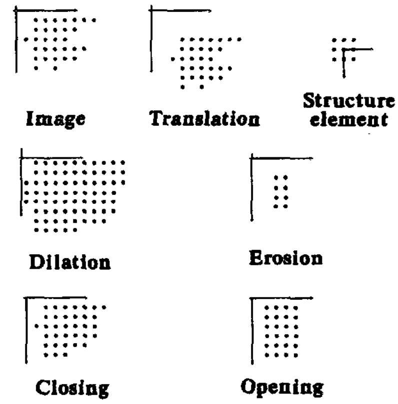


FIG. 3. Examples of basic morphological operations.

Definitions

Let X be a set representing a binary image. Let B denote a structuring element that describes a simple shape (e.g., square, circle). $(X)_y$ is defined as the translation of X by vector y, i.e., $(X)_y = \{x + y \mid x \in X\}$. Two fundamental morphological operations on X are defined as follows:

$$\text{erosion: } X \ominus B = \cap_{b \in B} (X)_{-b} \quad (1)$$

$$\text{dilation: } X \oplus B = \cap_{b \in B} (X)_{b} \quad (2)$$

Erosion is a shrinking of the original image and dilation is an expansion of the original image. In practice, dilations and erosions are usually employed consecutively: either an image is dilated then eroded or an image is eroded then dilated. In either case, iteratively applying dilations and erosions eliminates specific components smaller than the structuring element without any global geometric distortion.

The opening $\bigcirc$ and closing $\bullet$ operations are defined as

$$X \bigcirc B = (X \ominus B) \oplus B \quad (3)$$

$$X \bullet B = (X \oplus B) \ominus B. \quad (4)$$

Opening is anti-extensive and the closing is extensive; therefore, for any given structuring element B,

$$X \bigcirc B \subseteq X \quad (5)$$

$$X \bullet B \supseteq X \quad (6)$$

Figure 3 shows illustrations of the basic morphological operations (erosion, dilation, opening, and closing).

3. CHARACTER STRING EXTRACTION FROM OVERLAPPING TEXT/BACKGROUND IMAGES

Overlapping text/background images can be directly opened with an appropriate structuring element to remove the background components that touch character strings [17]. All connected components below the size of a given structure element are removed and the larger objects remain substantially unchanged. This approach to background removal is only suitable when the ratio of the width of text characters to the width of the background symbols is appropriately large. The word "we" was almost perfectly extracted from the overlapped background (Figs. 4a and 4b) by simply opening the image with the following equation:

$$X \bigcirc (B \oplus B) = (((X \ominus B) \ominus B) \oplus B) \oplus B \quad (7)$$

X is an image set, and B is a 3×3 square structuring element. However, a simple opening operation is not an effective approach to obtaining high quality text strings from text/background images which have a small ratio of the width of text characters to the width of background symbols. The results of opening an image set (Fig. 5a) with unsuitable structuring elements are shown in Figs. 5b and 5c.

The algorithm that we are presenting can extract character strings from regular periodic backgrounds regardless of the orientation and style of background symbols. The underlying strategy of our algorithm is to maximize background component removal while minimizing the shape distortion of text characters by using appropriate morphological operations. For proper segmentation, the overlap-

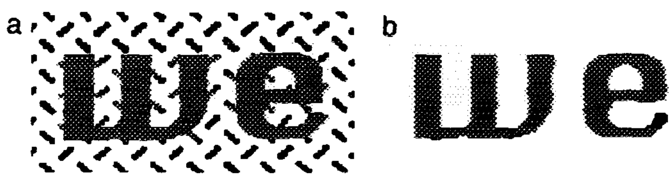


FIG. 4. (a) The word "we" with an uniform background. (b) Background removal after opening the image with a 9×9 square structuring element.

FIG. 5. (a) The original image. (b) The result opened with too small a structuring element. (c) The result opened with too large a structuring element.

ping text/background images should meet the following requirements:

(1) The graphic symbols in the background are periodically distributed.

(2) The width ratio of the minimal stroke of the character strings to the background symbols is approximate to 1.

(3) The resolution of the digitizer is not such that the topological property of each character is eliminated by low resolution.

However, acceptable segmentation results are still obtained even if some of these constraints are not strictly satisfied. Figure 6 shows a flow chart of our text/background segmentation system. The algorithm will be described in detail in the following sections.

3.1. *Parameter Estimation of Uniform Backgrounds*

By observing the structure of uniform backgrounds, we can easily see that the background symbols are periodically distributed in both the horizontal and vertical directions. Hence, it is possible to extract the background symbols from the text/background image when their "frequency" of repetition is known and proper structuring elements are chosen. For expressional simplicity, we use Periodic Distance (in pixel units) to describe the periodicity of the background symbols. The PDH and PDV represent the Periodic Distance in the Horizontal and Vertical directions respectively. They satisfy the following inequalities:

$$\mathrm{CL}((X \ominus T1) \ominus B_{\mathrm{PDH}}) > \mathrm{CL}((X \ominus T1) \ominus B_i) \quad i = 1, 2, \ldots, M; i \neq \mathrm{PDH} \tag{8}$$

$$\mathrm{CL}((X \ominus T2) \ominus B_{\mathrm{PDV}}) > \mathrm{CL}((X \ominus T2) \ominus B_j) \quad j = 1, 2, \ldots, M; j \neq \mathrm{PDV}. \tag{9}$$

X is the text/background image set, CL(E) is a pixel counting function that calculates the total number of image "1"

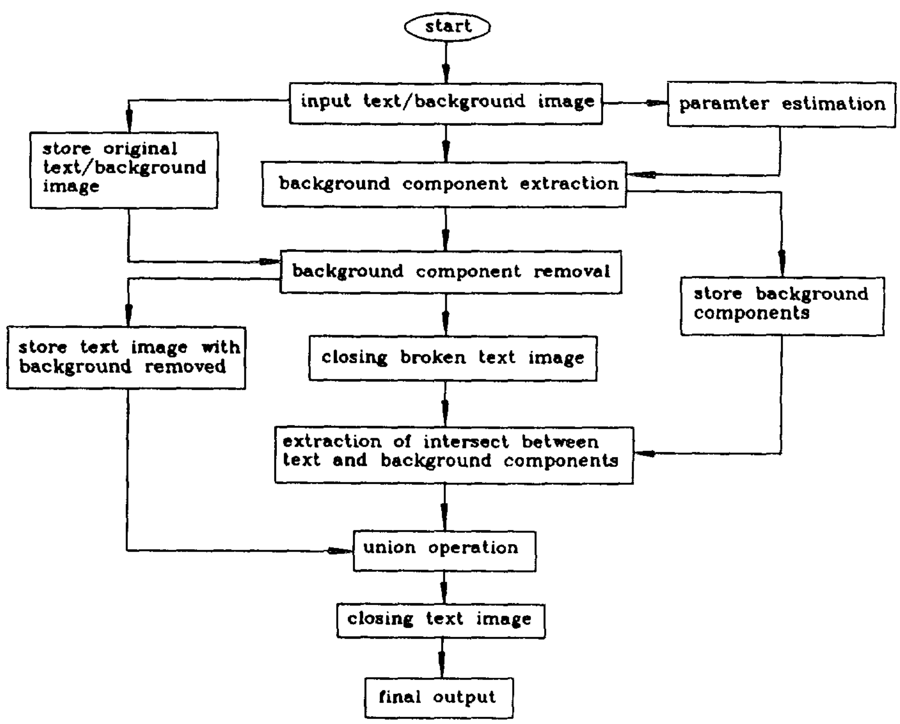


FIG. 6. A flow chart of the text extraction system.

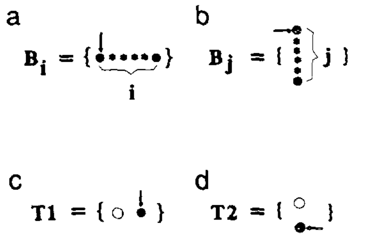


FIG. 7. (a), (b) Point pairs with i and j apart structuring elements B_i and B_j. (c), (d) The transit structuring elements $T1$ and $T2$. (●) Points of the structuring element belonging to binary image set X. (*) Don't care points. (↓), (←), (→), Locations of the origin associated with the structuring elements.

pixels for a given binary image set E, M is a constant, and B_i and B_j are two *point pair* structuring elements with i ($\leq M$) and j ($\leq M$) apart respectively as defined in Figs. 4a and 4b.

Two transit structuring elements, $T1$ and $T2$, defined in Fig. 7c and 7d, erode the image set X to extract the left and top edges; that is, only image pixels "1" which have blank pixels "0" in either their left or top sides are retained. The resultant left and top edges of Fig. 5a are shown in Figs. 8a and 8b. Because most redundant character and background pixels that may cause ambiguities and estimation errors have been removed, using the edge information of the characters and background components instead of all of the image pixels to estimate the PDH and PDV parameters produces accurate results. The procedure for calculating PDH and PDV is described in Appendix 1.

Figure 9 is a simple example of how the erosion operations are performed by the structuring element B_i (i = 1, 2, 3, 4) on a given edge image E. B_i (i = 1, 2, 3, 4) is applied respectively to erode the image E. The resulting images (Ei (i = 1, 2, 3, 4)) are shown in Figs. 9b–9e. ○ points and ● points in these figures denote the removed pixels and retained pixels which belong to the original image E. After erosion process is completed, the black pixel counting function is applied to the eroded images Ei (i = 1, 2, 3, 4) to calculate the total number of image pixels remaining in the eroded images. The black pixel counting procedures are simply expressed by CL(Ei) (i = 1, 2, 3, 4). For example, after the structure element B_1 erodes the image E, there remain 4 black image pixels (Fig. 9b). As seen in Figs. 9b–9e, CL(Ei) is maximized

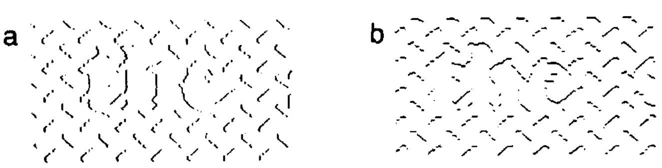


FIG. 8. (a) The left edges of the text/background components of Fig. 5a. (b) The top edges of the text/background components of Fig. 5a.

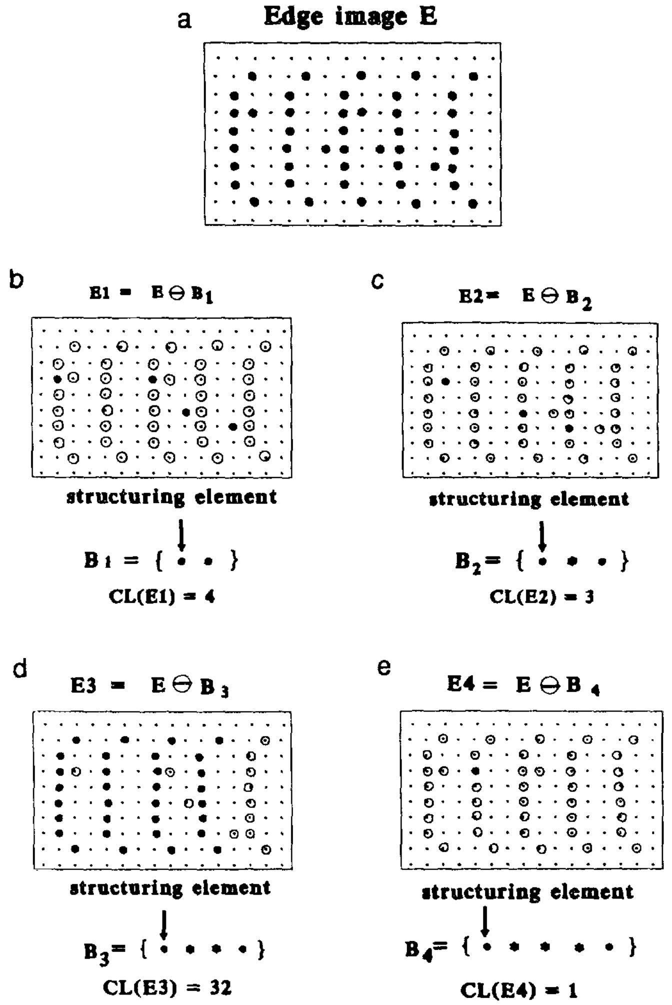


FIG. 9. (a) An initial edge image E generated for illustration. From (b) to (e) the symbols concerning the structuring elements B_i, i = 1, 2, 3, 4. (●) Points of the image which belong to Ei (i = 1, 2, 3, 4). (○) Points of the image which belong to Ei^c (i = 1, 2, 3, 4). (·) Points of the image which indicate pixel positions. (↓) Locations of the origin of the structuring elements. (*) Don't care. ($CL(X)$) The pixel counting function on a given image set X.

with i = 3: after application of B_3, CL($E3$) = 32 (see Fig. 9d). According to Eq. (8), the subscript index 3 of the structuring element B_3 represents the PDH value of the edge image E. Similarly, the PDH and the PDV are obtained (Figure 10) based on the edge images Figs. 8a and 8b, and are used in the next step to extract the background symbols from the text/background image.

3.2. *Morphological Operations for Extracting Background Components*

A morphological process is designed to extract the background symbols or remove the character strings from a specific text/background image. Based on the parameters PDH and PDV obtained from Eq. (8) and Eq. (9), four structuring elements, $S1$, $S2$, $S3$, and $S4$, are designed to

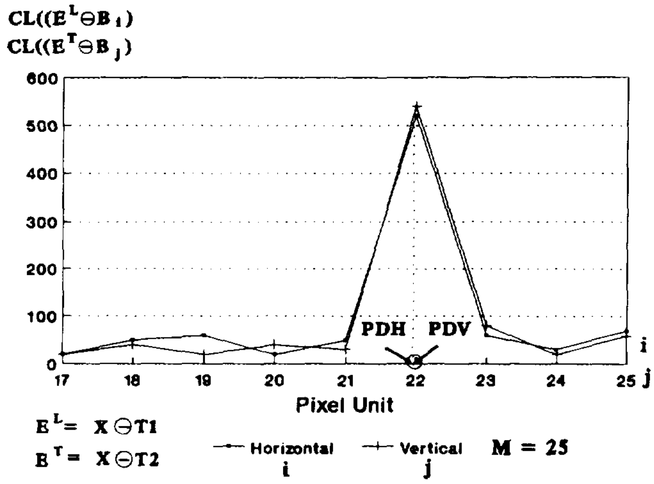


FIG. 10. Histograms of the $CL(B^L \ominus B_i)$ and $CL(B^T \ominus B_j)$. E^L and E^T are the left edges and top edges. $E^L = X \ominus T1$, $E^T = X \ominus T2$, X is a text/background image set. PDH and PDV indicate values corresponding to the maximums of the histograms in the horizontal and vertical directions.

realize the appropriate morphological erosion operations (Fig. 11). These erosion operations are used to remove the text strings from the text/background images. For example, an erosion operation with the structuring element $S1$ allows a scanned image pixel "1" to be removed only if the pixels PDH − 1, PDH, and PDH + 1 pixel units to the right of the original pixels are all "0" pixels. Otherwise, the scanned image pixel "1" is unchanged. Using three reference points in the structuring elements to determine the binary value of image pixels helps to prevent the erosion operations from excessively eliminating background pixels that have been distorted by optical devices such as digitizers, laser printers, and etc.

The recursive erosion $\mathrm{RE}_i(X)_S$ of an object image X with the structuring element S is defined by the following relation:

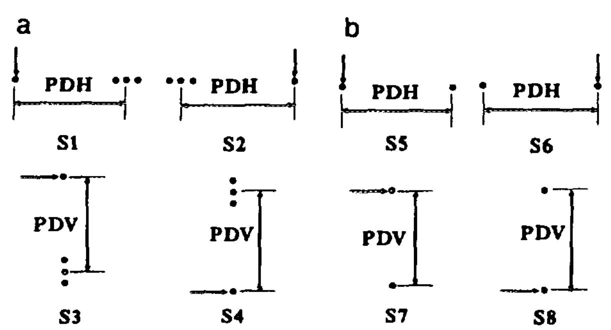


FIG. 11. (a) Structuring elements ($S1$, $S2$, $S3$, $S4$). (b) Structuring elements ($S5$, $S6$, $S7$, $S8$). (●) Points of the structuring element which must belong to image set X. (↓) Locations of the origin of the structuring elements.

$$\begin{aligned}\mathrm{RE}_i(X)_S &= (X \ominus S)_i \\ &= ((\ldots((X \ominus S) \ominus S)\ldots \ominus S))\; i\,\text{time}\end{aligned} \tag{10}$$

therefore, the whole procedure can be described by the recursive relation

$$\mathrm{Re}_i(X)_{(S1|S2|S3|S4)} = (((((X \underset{(LR)}{\ominus} S1) \underset{(TB)}{\ominus} S3) \underset{(RL)}{\ominus} S2) \underset{(BT)}{\ominus} S4)))_i. \tag{11}$$

$S1$ and $S3$ erode the image X from left to right (LR) and from top to bottom (TB) respectively. Similarly, $S2$ and $S4$ erode the image X from right to left (RL) and bottom to top (BT) respectively. This alternative multidirectional scan increases the convergence rate and ensures that all possible image pixels that do not belong to the background are replaced with blank pixels.

The erosion procedure is an iterative erosion of the image X until the termination condition

$$X_i = X_{(i-1)} \ominus Sj, \qquad j = 1, 2, 3 \text{ and } 4 \tag{12}$$

is satisfied, X_i and X_{i-1} denote the current eroded image and the previous eroded image respectively. When the termination condition (12) is satisfied, it also means that $\mathrm{CL}(X_i) - \mathrm{CL}(X_{i-1}) < \alpha$. α is a small constant and $\mathrm{CL}(X)$ is the pixel counting function applied to the image set X. Figure 12 shows the results of the background extraction process after 9 iterations of the erosion operation with the structuring elements $S1$, $S2$, $S3$, and $S4$. Figure 13 shows the results of the background extraction process after 4 iterations of the erosion operation with the *point pair* structure elements $S5$, $S6$, $S7$, and $S8$. Comparing Figs. 12 and 13, although the convergency rate of the background extraction process is faster with structuring elements Si (i = 5, 6, 7, 8) than that with structuring elements Si (i = 1, 2, 3, 4), the significant loss of the background pixels can be obviously observed in Fig. 13d. The loss of background pixels will produce noise on text images with background removed. The extra noise filtering will degrade the quality of restoration of text parts.

The procedure for extracting background pixels is described in Appendix 2.

3.3. *Extraction of Character Strings from the Background*

After the background extraction procedure, the background removal operation (exclusive-OR (XOR) operation)

$$Z = \mathrm{XOR}(X, Y) \tag{13}$$

is performed on the original text/background image set X (Fig. 14a) and the background image set Y (Fig. 14b).

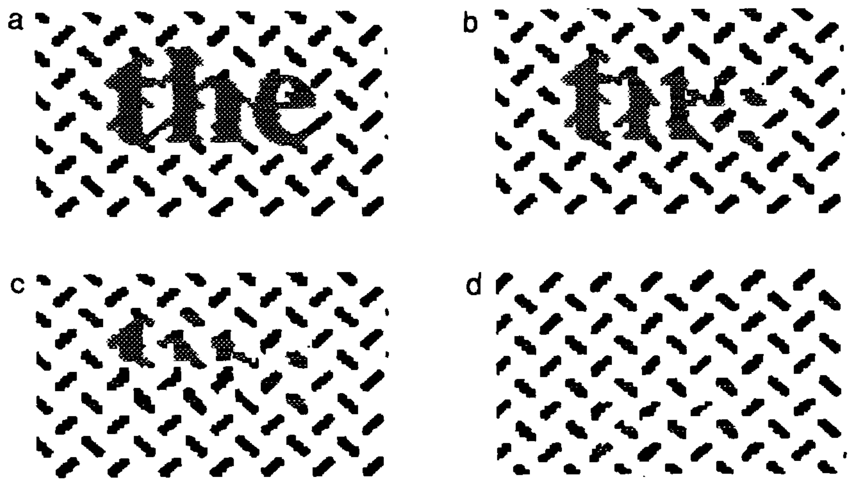


FIG. 12. (a) The original text/background image. (b) The image eroded by $S1$ after one iteration. (c) The image eroded by $S2$ after two iterations. (d) the background extracted after nine iterations.

The result Z denotes the character string image with background removed (Fig. 14c). Directly applying closing operation on the image Z (i.e., $Z \bullet B$, where B is an appropriate structuring element) fills the internal gaps of the character strings. However, this produces serious shape distortion. To improve the results of the morphological operation, a conditional dilation is performed before the closing operation:

$$W = ((Z \oplus B1) \cap Y) \bullet B2 \tag{14}$$

$B1$ and $B2$ are two suitable structuring elements. The size of these structuring elements depends largely on the resolution of the image and the size of the background symbols.

Dilation along the background pixels, i.e., $D = (Z \oplus B1) \cap Y$ (Fig. 14d), compensates for the pixels lost during background removal. Figure 14e shows the image after application of Eq. (14). However, Eq. (14) still causes a loss of image detail even though it preserves most topological properties of the character strings.

The final operation for character string extraction is

$$R = ((W \cap Y) \cup Z) \bullet B3 \tag{15}$$

Here $B3$ is a proper structuring element that is used to fill the narrow crack between the union components ($U = W \cap Y$, in Fig. 14f) and the characters with internal gaps (Fig. 14c). R represents the final image (Fig. 14h).

Those text character pixels that do not overlap with background components should remain unchanged if the character shape distortion is to be minimized. The operation described by Eq. (15) can be effectively applied to

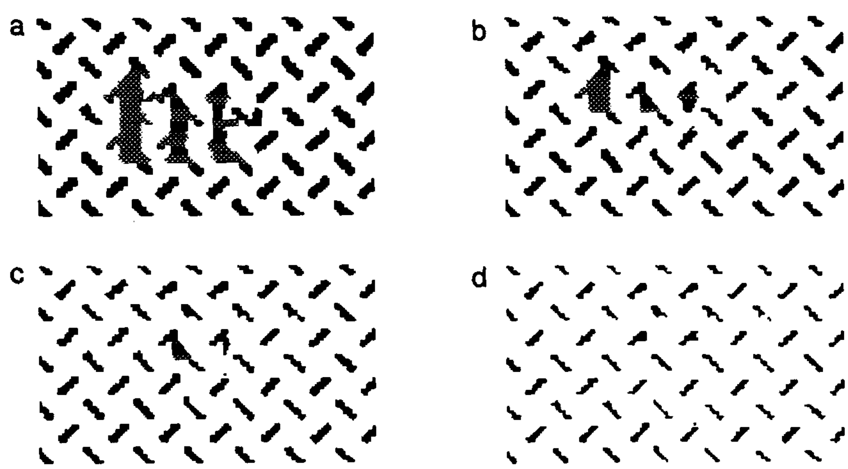


FIG. 13. (a) The image eroded by $S5$ after one iterations. (b) The image eroded by $S6$ after two iterations. (c) The image eroded by $S7$ after three iterations. (d) The image eroded by $S8$ after four iterations.

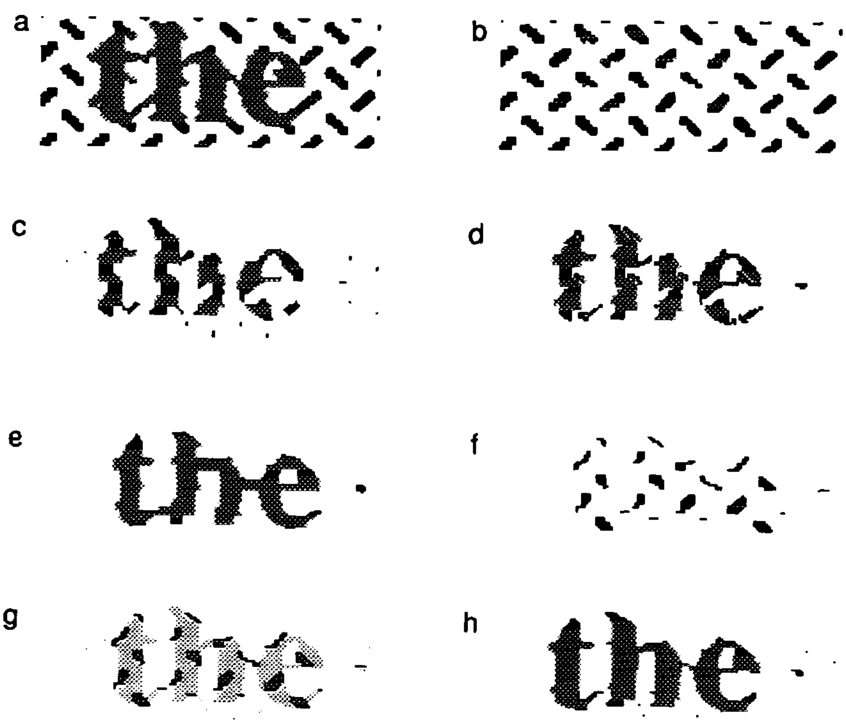


FIG. 14. (a) The original overlapped text/background image (X). (b) The extracted background components (Y). (c) The image after background removal (Z). (d) The characters obtained by a conditional dilation operation ($D = (Z \oplus B1) \cap Y$)). (e) The characters by closing operation ($W = D \cdot B2$). (f) The union parts of (b) and (e) ($U = W \cap Y$). (g) The characters after ORing (f) and (c) ($U \cup Z$). (h) The final result ($R = ((W \cap Y) \cup Z) \cdot B3$).

improve the performance of the character extraction (Fig. 14h, compare Fig. 14e).

The procedure for text restoration is described in Appendix 3.

4. EXPERIMENTAL RESULTS

Several test image data are obtained by a scanner (ScanMaker II) which interfaced with an Sun SPARC 10 workstation to test and evaluate the performance of the algorithm. For these experiments, six artificial images generated by Macintosh software (Canvas 3.00) and one headline image from a Chinese newspaper are used to test the proposed algorithm. The system performs no character recognition and only extracts the character strings from the overlapping text/background images. If an image conforms to the constraints mentioned in Section 3, the performance of the algorithm is effective. Figures 15 and 16 show the illustrative experimental results. These images demonstrate the algorithm's ability to extract text strings from overlapping text/background images with acceptable shape distortion. However, some shape distortion still occurred in our tests due to the high degree of overlap and small width ratio of the text strings to background components (see Figs. 15 and 16). In addition, if the practical headline image does not have uniformly distributed background in the whole images (see Fig. 16(a)), the modified headline image (Fig. 16(b)) can be easily generated by using the algorithm given in Appendix 4.

As observed in our experiments, the morphological operations which are quite useful for image processing and analysis require a great deal of computation to implement. Fortunately, many commercial special-purpose image processors utilizing parallelism have been developed in recent years [5, 11, 18, 19]. In addition, there are VLSI techniques [20] for implementing the combinations of erosions and dilations. These available techniques make it possible to implement these operations on large images in real time.

5. CONCLUSION

A new algorithm for text string extraction from overlapping text/background images has been developed. The performance of the new scheme was tested on six artificial images and one real image, each with a different style of periodically distributed background symbols. Provided that the images conformed to several constraints, the algorithm was effective and reliable. This new method is appropriate for implementation in document analysis systems. We focus only on the extraction of characters from

410

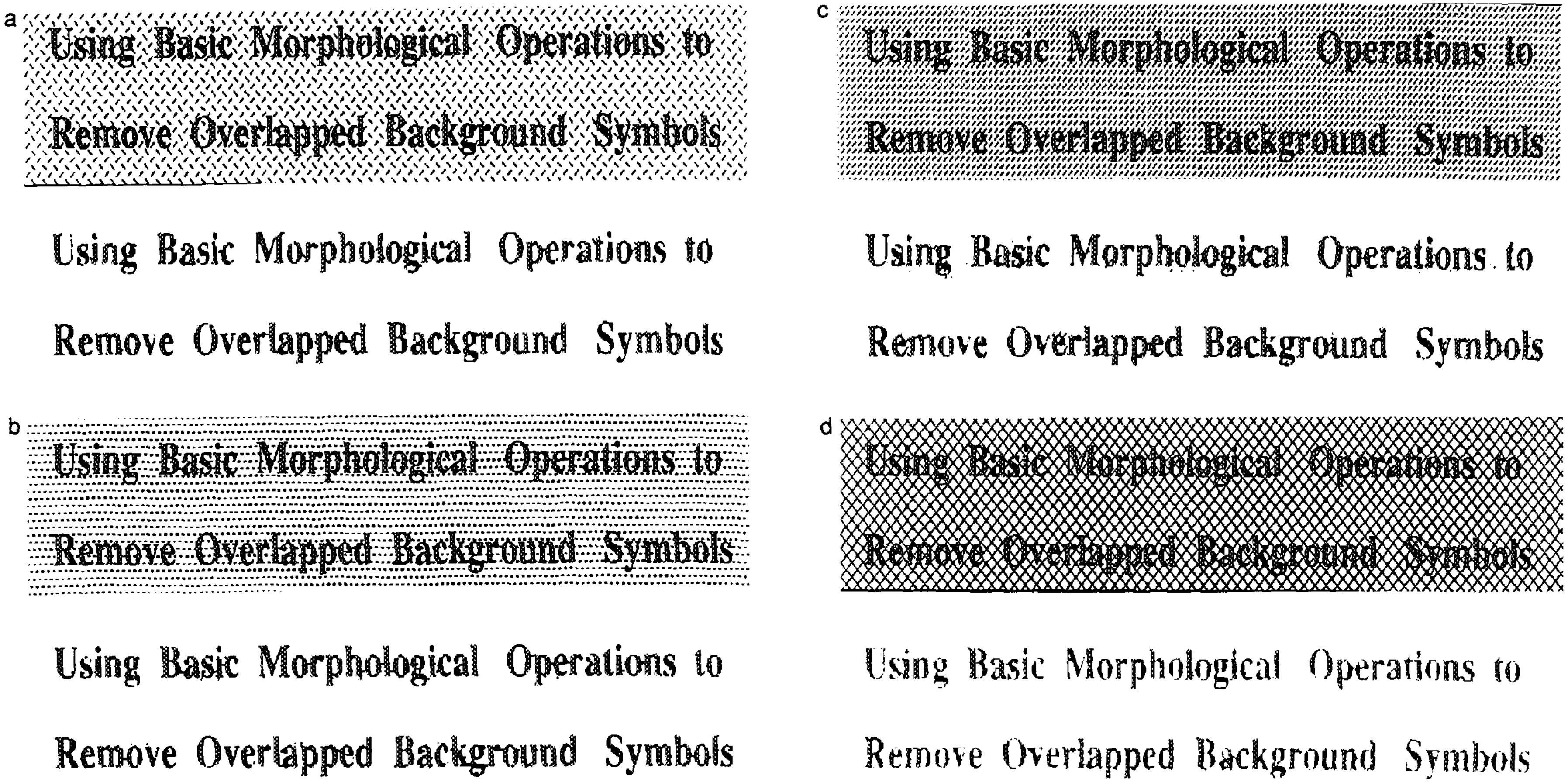


FIG. 15. The character images extracted from different text/background images.

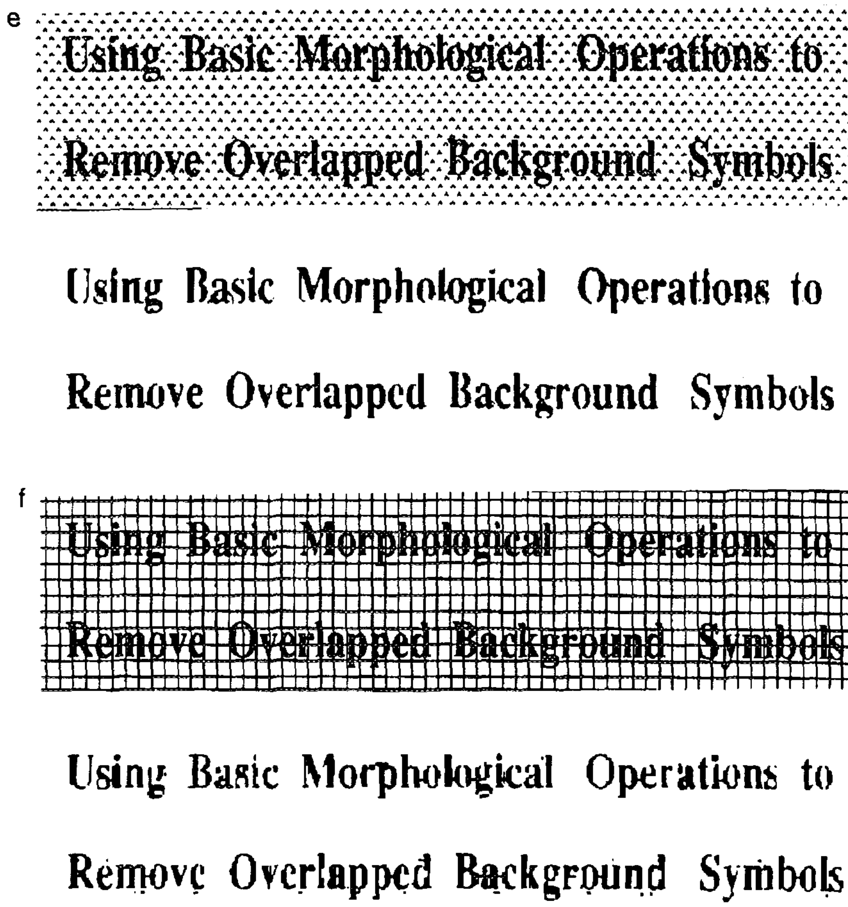


FIG. 15—*Continued*

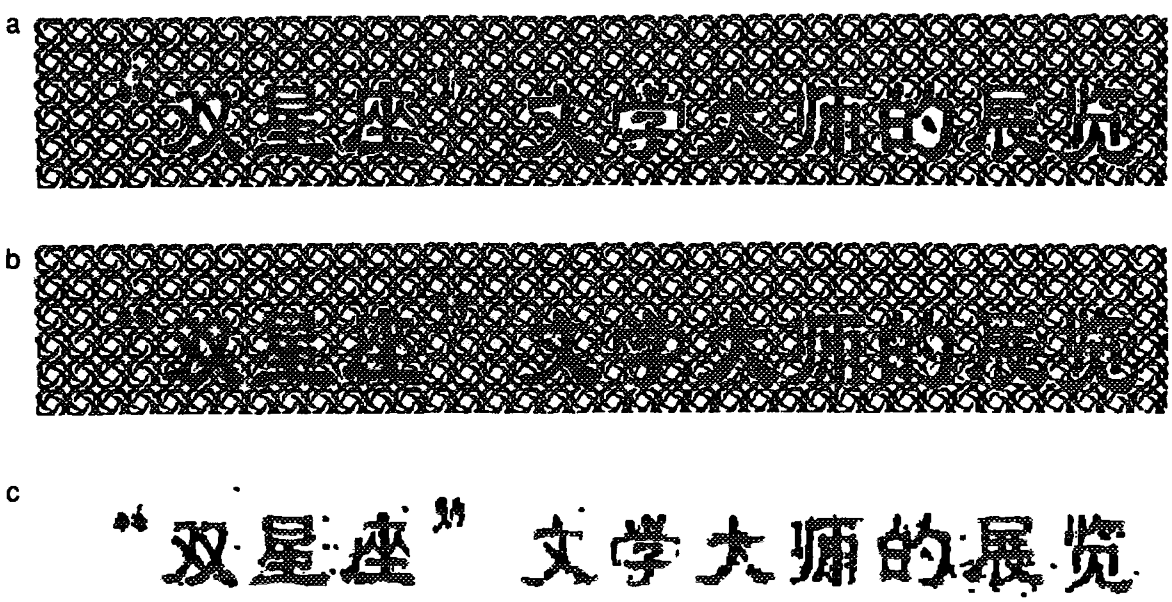


FIG. 16. The character images extracted from a text/background headline image scanned from Chinese newspaper. (a) Original headline image. (b) Modified headline image. (c) Final text extraction.

the regular periodic overlapping backgrounds in this paper, since these kinds of background symbols can be easily generated by computer publishing devices.

The uniform backgrounds can be considered as one sort of texture. However, most texture analysis techniques, such as statistical approaches, structural approaches, and spectral approaches [6], are effective and suitable for the gray level images. For binarized document images, our algorithm based on binary mathematical morphology would be very effective.

APPENDIX 1

Procedure for Calculating PDH and PDV Described in *C*-like Format

Procedure for left edges extraction:

```
for( r from 1 to Image_Row ; c from 1 to Image_Col) {
    image_left_edge[r][c] = image_text_background[r][c] ⊖ T1;
                                                        }
```

Procedure for top edges extraction:

```
for( r from 1 to Image_Row ; c from 1 to Image_Col) {
    image_left_edge[r][c] = image_text_background[r][c] ⊖ T2;
                                                        }
```

Procedure for calculating PDH:

```
for( k from 1 to M){
   for( r from 1 to Image_Row ; c from 1 to Image_Col) {
      imagebuff_k [r][c] = image_left_edge[r][c] ⊖ B_i ;
        N_k-1 = CL(imagebuff_k-1 );
        N_k = CL(imagebuff_k );
        N_k+1 = CL(imagebuff_k+1 );
          if( |N_k - N_k-1| > threshold && |N_k - N_k+1| > threshold)
              PDH = k ;                                  }
           }
```

Procedure for calculating PDV:

```
for( k from 1 to M){
   for( c from 1 to Image_Col; r from 1 to Image_Row ;) {
      imagebuff_k [r][c] = image_left_edge[r][c] ⊖ B_j ;
        N_k-1 = CL(imagebuff_k-1 );
        N_k = CL(imagebuff_k );
        N_k+1 = CL(imagebuff_k+1 );
          if( |N_k - N_k-1| > threshold && |N_k - N_k+1| > threshold)
              PDH = k ;                                  }
           }
```

APPENDIX 2

Procedure for Extracting Background Pixels Described in *C*-like Format

```
     image_background_1[r][c] = original_image[r][c];

for( i from 1 to Number_iteration)
{
     for( r from 1 to Image_Row ; c from 1 to Image_Col) {
        image_background_i [r][c] = image_background_i-1 [r][c] ⊖ S1; }
           if(image_background_i-1 [r][c]-image_background_i[r][c] < threshold)
              procedure_end;
     for( r from 1 to Image_Row ; c from Image_Col to 1) {
        image_background_i [r][c] = image_background_i-1 [r][c] ⊖ S2; }
          if(image_background_i-1[r][c]-image_background_i[r][c] < threshold)
             procedure_end;
     for( r from 1 to Image_Row; c from 1 to Image_Col) {
         image_background_i [r][c] = image_background_i-1 [r][c] ⊖ S3; }
           if(image_background_i-1 [r][c]-image_background_i [r][c] < threshold)
             procedure_end;
     for( r from Image_Row to 1 ; c from 1 to Image_Col ) {
         image_background_i [r][c] = image_background_i-1 [r][c] ⊖ S4; }
           if(image_background_i-1 [r][c]-image_background_i [r][c] < threshold)
             procedure_end;

}
```

APPENDIX 3

Procedure for Text Restoration Described in *C*-like Format

```
for( r from 1 to Image_Row ; c from 1 to Image_Col)
 {
   if(image_text_background[r][c] == 1)
   image_background_removal[r][c] =
      image_text_background[r][c] XOR image_background[r][c];
 }

for( r from 1 to Image_Row; c from 1 to Image_Col)
   do  {
  image_background_removal[r][c]
          = removal_isolate_pixels(image_background_removal);
       }
   do  {
  image_text1[r][c] = closing1(image_background_removal);
       }
   do  {
  image_intersect[r][c] = intersect(image_text1,image_background);
       }
   do  {
  image_text2[r][c] = union(image_backgd_removal,image_background_removal);
       }
   do  {
  image_text_final[r][c] = closing2(image_text2);
       }
```

APPENDIX 4

Procedure for Modification of Headline Images Described in *C*-like Format

```
for( r from 1 to Image_Row ; c from 1 to Image_Col)
 {
   image_headline_modified[r][c] = image_headline[r][c];
 }

for( r from 1 to PDV; c from 1 to Image_Col)
  {
     k = PDV; + r;
while( k < Image_Row && k >= PDV){
  if( image_headline[r][c]==1 && image_headline[k][c]==0)
      image_headline_modified[k][c] = 1;
        k = k + PDV;
                                 }
  }
```

REFERENCES

1. J. Schurmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler, and M. Oberlander, Document analysis—Form pixels to contents, *IEEE Proc.* **80**(7), 1992, 1101–1119.
2. H. Ozawa and T. Nakagawa, A character image enhancement method from characters with various background images, in *Proceedings, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993*, pp. 58–61.
3. S. H. Joseph, On the extraction of text connected to linework in document images in *Proceedings, First International Conference on Document Analysis and Recognition, Saint-Molo, France, Sept. 30–Oct. 2, 1991*, Vol. 2, pp. 993–999.
4. N. Bilawala, P. Hart, and M. Peairs, Image Continuation in *Proceedings, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993*, pp. 53–57.
5. R. M. Lougheed, An overview of grayscale morphological filters in *23th Asilomar Conference on signals, Systems, and Computers, Oct.30–Nov. 1, Pacific Grove, California, 1989*, Vol. 1, pp. 152–156.

6. H. Yamada, K. Yamamoto, T. Saito, K. Hosokawaa, and H. Yanagisawa, Laser-marked alphanumeric character recognition by multi-angled matching method, in *11th IAPR International Conference on Pattern Recognition, The Hague, The Netherlands, Aug. 30–Sept. 3, 1992*, pp. 326–329.
7. J. M. White and G. D. Rohrer, Image thresholding for optical character recognition and other applications requiring character image extraction, *IBM J. Res. Dev.* **27**(4), 1983, 400–410.
8. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison–Wesley, 1992.
9. J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognit.* **29**(1), 1986, 41–47.
10. Y. Liu, R. Fenrich, and S. N. Shrihari, An object attribute threshold algorithm for document image binarization, *Proceedings, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993*, pp. 278–281.
11. P. Maragos, A representation theory for morphological image and signal processing, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(6), 1989, 586–599.
12. R. M. Haralick, S. R. Sternberg, and X. Zhuang, Image analysis using mathematical morphology *IEEE Trans. on Pattern Anal. Mach. Intell.* **9**(4), 1987, 532–550.
13. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
14. J. Serra, Introduction to mathematical morphology *Comput. Vision, Graphics, Image Process.* **35**, 1986, 283–305.
15. Robert C. Vogt, *Automatic Generation of Morphological Set Recognition Algorithms* Springer-Verlag, New York/Berlin, 1989.
16. I. Pitas and A. N. Venetsanopoulos, Morphological shape representation, *Pattern Recognition.* **25**(6), 1992, 555–565.
17. C. R. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
18. M. J. B. Duff, D. M. Watson, T. M. Fountain, and G. K. Shaw, A cellular logic array for image processing, *Pattern Recognit.* **5**, 1973, 229–247.
19. F. A. Gerritsen and L. G. Aardema, Design of use of DIP-1: A fast flexible and dynamically microprogrammable image processor, *Pattern Recognit.* **27**, 1984, 115–123.
20. F. A. Gerritsen and P. W. Verbeek, Implementation of cellular logic operation using 3 × 3 convolution and table lookup hardware, *Comput. Vision, Graphics, Image Process.* **27**, 1984, 319–330.