CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 55, No. 3, May, pp. 203–217, 1993

Extraction of Binary Character/Graphics Images from Grayscale Document Images

MOHAMED KAMEL[1] AND AIGUO ZHAO[2]

Pattern Analysis and Machine Intelligence Laboratory, Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Received November 27, 1991; revised October 16, 1992; accepted December 14, 1992

The extraction of binary character/graphics images from grayscale document images with background pictures, shadows, highlight, smear, and smudge is a common critical image processing operation, particularly for document image analysis, optical character recognition, check image processing, image transmission, and videoconferencing. After a brief review of previous work with emphasis on five published extraction techniques, viz., a global thresholding technique, YDH technique, a nonlinear adaptive technique, an integrated function technique, and a local contrast technique, this paper presents two new extraction techniques: a logical level technique and a mask-based subtraction technique. With experiments on images of a typical check and a poor-quality text document, this paper systematically evaluates and analyses both new and published techniques with respect to six aspects, viz., speed, memory requirement, stroke width restriction, parameter number, parameter setting, and human subjective evaluation of result images. Experiments and evaluations have shown that one new technique is superior to the rest, suggesting its suitability for high-speed low-cost applications. © 1993 Academic Press, Inc.

1. INTRODUCTION

Documents are the primary information medium in the information society today. Common symbols representing information on documents are handwritten or machine-printed characters and graphics. With the availability of computers, video terminals, optical disks, fax machines, laser printers and scanners, these symbols can be generated, transmitted, accessed, stored, displayed, printed out and reabsorbed automatically. As a substitute for manual handling, document image analysis systems can recognize the symbols on documents and organize them in either data files which can be manipulated by various word processing and graphics softwares or databases. Various such systems have recently appeared or proposed in both academic journals and commercial markets. The systems range from PC-based OCR (Optical Character Recognition) software packages [1] to special systems which include automatic character recognition and graphics vectorization systems for engineering drawings [2, 3], maps [4], and automatic mail sorting systems [5, 6].

The application of these systems is greatly limited by their requirements for good quality documents. However, clean good quality documents are seldomly encountered in real applications. Some common problems which contribute to poor-quality documents are: (1) highlighted characters in various colours, (2) smeared or smudged characters/graphics, (3) nonuniform change in colours due to long term storage, (4) poor writing or printing quality, and (5) shadows due to poor lighting conditions when the document image is captured. Although more processing can be made in later phases, these problems are mainly handled in the first phase of a document analysis system. In this phase, a grayscale image is first captured and digitized using a scanner or a video camera, then a binary image is extracted from the original grayscale image using certain extraction technique. These poor-quality documents are also often handled by videoconferencing systems which transmit document images for human recognition. It is sufficient to represent a character/graphics image in a binary image which will be more efficient to transmit and process instead of the original grayscale image. Although often in good quality, checks provide us a similar problem. In the perspective electonic check image processing systems which are being developed by some manufacturers, once a check arrives, its image is transmitted, stored and accessed electronically, and the characters on it will be automatically recognized using OCR readers. At the first phase of such systems, a binary character image is extracted from the original grayscale check image and the background picture is suppressed as much as possible. Although sophisticated extraction techniques have been published [3, 7–9], our experiments have shown that further effort to solve these problems is still necessary.

This paper is concerned with the extraction of binary

[1] To whom correspondence should be addressed: (519)888-4645 (phone) and/or mkamel@watnow.uwaterloo.ca (e-mail).

[2] Mr. Zhao is now with Atlantis Scientific Systems Group Inc., Ottawa, Canada.

1049-9652/93 $5.00

character/graphics images from grayscale document images. In Section 2, previous work with an emphasis on five published techniques is briefly reviewed. In Section 3, two new techniques are presented. In Section 4, speed, memory requirement, stroke width restriction, parameter number, parameter setting, and human subjective evaluation of result images are proposed and discussed as six aspects for evaluating and analyzing extraction techniques. With experiments on a typical check image (Fig. 1) and a common text document image with highlighted characters and shadows (Fig. 2), both new and published techniques are evaluated and analysed with respect to these six aspects. The last section includes a summary and conclusion drawn from this work.

2. REVIEW OF PREVIOUS WORK

One of the earliest but also most simple techniques for extracting a binary image from an original grayscale image is the global thresholding technique, which can be found in text books on image processing, see, for example, [10]. This technique does not consider the difference between character/graphics and background, and therefore usually cannot give satisfactory results. Later techniques consider the difference between character/graphics and background. They consider both characters and graphics as line drawings and either fully or partially enforce a restriction on stroke width by assuming that the stroke widths of all these line drawings lie in a predetermined range. Based on this restriction, every character/graphics pixel is extracted according to not only its own gray level, but also the gray levels of the pixels in some approximate character-size neighborhood around the pixel. These techniques are called locally adaptive techniques [7, 11]. However, these techniques do not differentiate characters from graphics and treat the separation of characters from graphics as another problem for which some techniques have been reported in [12–16]. In this paper, we also follow this approach, and therefore will not differentiate characters from graphics.

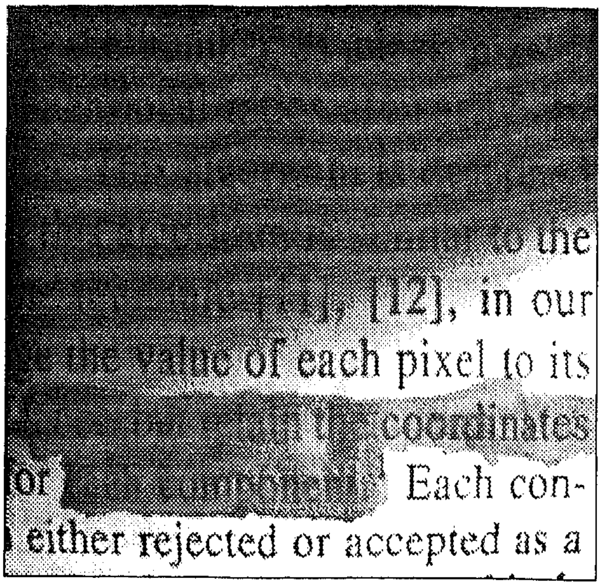

[12], in our
the value of each pixel to its
coordinates
Each con-
either rejected or accepted as a

FIG. 2. A 512 × 480 × 8 original document image with highlighted characters and shadows.

A survey on initial work in this field can be found in [17]. Recent techniques originating from various application areas including check image processing [8, 18], OCR [11], blueprint image extraction [3], and videoconferencing systems [9] have been reported. An evaluation of three such techniques viz., the nonlinear function technique [8], the local contrast technique [11], and a second derivative technique which is virtually equivalent to the integrated function technique in [8], has been presented by Palumbo, Swaminathan, and Srihari [7].

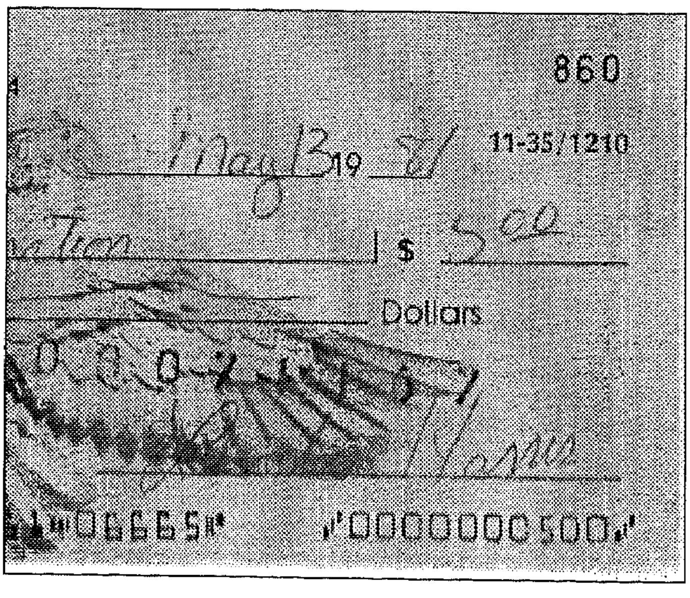


FIG. 1. A 640 × 512 × 8 original check image.

Considered as part of the optical scanning process, this extraction is often implemented using minimal software and/or hardware in scanners or personal computers. Therefore, we put our research emphasis on the extraction techniques which have relative high-speed performances and minimal memory requirements, and therefore are suitable for high-speed low-cost applications. In this section, we briefly review five published techniques which will be evaluated and compared with our two new techniques later. For simplicity, unless specified, the physical meaning of every mathematical symbol introduced from now on will not be changed.

2.1. *Global Thresholding Technique*

Let $f(x, y)$ with $1 \le x \le M$, $1 \le y \le N$, be the original grayscale image with width of M, height of N and gray level range of [0, 255], $b(x, y)$ be the extracted binary character/graphics image with gray level 1 representing character/graphics and 0 representing background, and T be a predetermined constant, then the mathematical description of this technique is

$$b(x, y) = \begin{cases} 1 & \text{if } f(x, y) \le T \\ 0 & \text{otherwise.} \end{cases}$$

2.2. *Technique by Yasuda, Dubios, and Huang (YDH Technique)*

This technique was proposed as the first phase of a check image coding scheme by Yasuda, Dubois, and Huang [18]. It uses a combination of fundamental image processing techniques which can be mathematically described as follows:

1. *Dynamic Range Expansion (normalization)*

$$f_1(x, y) = \frac{f(x, y) - min}{max - min}$$

where

$$max = \max_{1 \le x \le M, 1 \le y \le N} [f(x, y)];$$

$$min = \min_{1 \le x \le M, 1 \le y \le N} [f(x, y)].$$

2. Smoothing.

$$f_2(x, y) = \begin{cases} f_1(x, y) & \text{if } range(x, y) > T_1 \\ \sum_{(x', y') \in A(x, y)} f_1(x', y')/8 & \text{otherwise} \end{cases}$$

where

$$range(x, y) = \max_{(x', y') \in A(x, y)} f_1(x', y') - \min_{(x', y') \in A(x, y)} f_1(x', y');$$

$$A(x, y) = \{(x', y') \| x' - x| \le 1, |y' - y| \le 1 \text{ and } (x', y') \ne (x, y)\};$$

T_1 is a predetermined parameter.

3. *Adaptive Thresholding and Dynamic Range Expansion.* Dividing the whole image area into overlapping blocks with size of $T_2 \times T_2$, for any (x, y) in the i^{th} block, we have

$$f_3(x, y) = \begin{cases} 255 & \text{if } maxb(i) - minb(i) < T_3 \text{ or } f_2(x, y) \ge aveb(i) \\ \dfrac{f_2(x, y) - minb(i)}{aveb(i) - minb(i)} 255 & \text{otherwise,} \end{cases}$$

where

$$maxb(i) = \max_{(x, y) \in i\text{th block}} [f_2(x, y)];$$

$$minb(i) = \min_{(x, y) \in i\text{th block}} [f_2(x, y)];$$

$$aveb(i) = \sum_{(x, y) \in i\text{th block}} [f_2(x, y)]/(T_2 \times T_2);$$

T_2, T_3 are predetermined parameters.

4. *Segmentation.*

$$b(x, y) = \begin{cases} 1 & \text{if } min3(x, y) < T_4 \text{ or } \sigma(x, y) > T_5 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$min3(x, y) = \min_{-1 \le i \le 1, -1 \le j \le 1} [f_3(x - i, y - j)];$$

$$\sigma(x, y) = \left(\sum_{-1 \le i \le 1, -1 \le j \le 1} [f_3(x - i, y - j) - ave3(x, y)]^2 \right)^{1/2} \Big/ 3;$$

$$ave3(x, y) = \sum_{-1 \le i \le 1, -1 \le j \le 1} [f_3(x - i, y - j)]/9;$$

T_4, T_5 are predetermined parameters.

2.3. *Nonlinear Adaptive Technique*

This technique as proposed in [8], conceptually compares the gray level of every pixel with some average of gray levels in a neighborhood, about the pixel, whose size is approximately equal to the character-size. This average is calculated with minimal memory and high-speed using two nonlinear equations. After adjusted by a bias function, the gray level of the processed pixel is compared with the calculated average. The mathematical description of this technique is as follows:

$$f_H(x, y) = f_H(x - 1, y) + W_H[f_H(x - 1, y) - f(x, y)]$$

$$f_V(x, y) = f_V(x, y - 1) + W_V[f_V(x, y - 1) - f_H(x, y)]$$

$$b(x, y) = \begin{cases} 1 & \text{if } Z_b[f(x - L, y)] > f_V(x, y) \\ 0 & \text{otherwise,} \end{cases}$$

where

W_H, W_V are two predetermined update functions. Either W_H or W_V has a value of zero only when its argument has a value of zero, and otherwise has

a value between zero and the value of its argument;
Z_b is a predetermined bias function;
L is a "look ahead" factor;
$f_H(x, y)$ is the horizontal average calculated from the gray levels of pixels at all the points (x', y) such that $1 \leq x' \leq x$;
$f_V(x, y)$ is the local average calculated from the gray levels of pixels at all the points (x', y') such that $1 \leq x' \leq x$ and $1 \leq y' \leq y$.

Figure 3 gives the update functions and the bias function adopted by White and Rohrer [8]. The update function provides the rate of response of the average to changes in the gray levels. Its value is zero only when its argument is zero, otherwise it has a value between zero and the value of its argument. This will guarantee that the average never exceeds the range of the input and will converge to the input for uniform gray areas. The bias function is used to offset the decision level and to eliminate the noisy background.

The major disadvantage of this technique is that it creates some black shadows in the background and extracts the left-top part edges of the large dark areas. Figures 4 and 5 illustrate this disadvantage by the black shadows in the check image and the left-top part edge of the highlight dark area in the text document image. One reason for this disadvantage is that the local average $f_V(x, y)$ is calculated based on the pixels in the left-top direction of the processed pixel only. Therefore a possible improvement can be made as follows. After extracting a binary image using the nonlinear adaptive technique, another binary

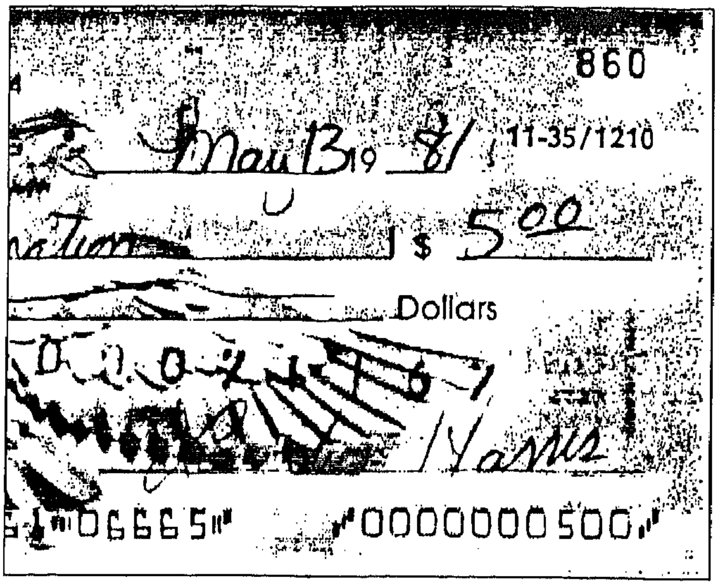


FIG. 4. Binary check character image extracted using Nonlinear Adaptive Technique.

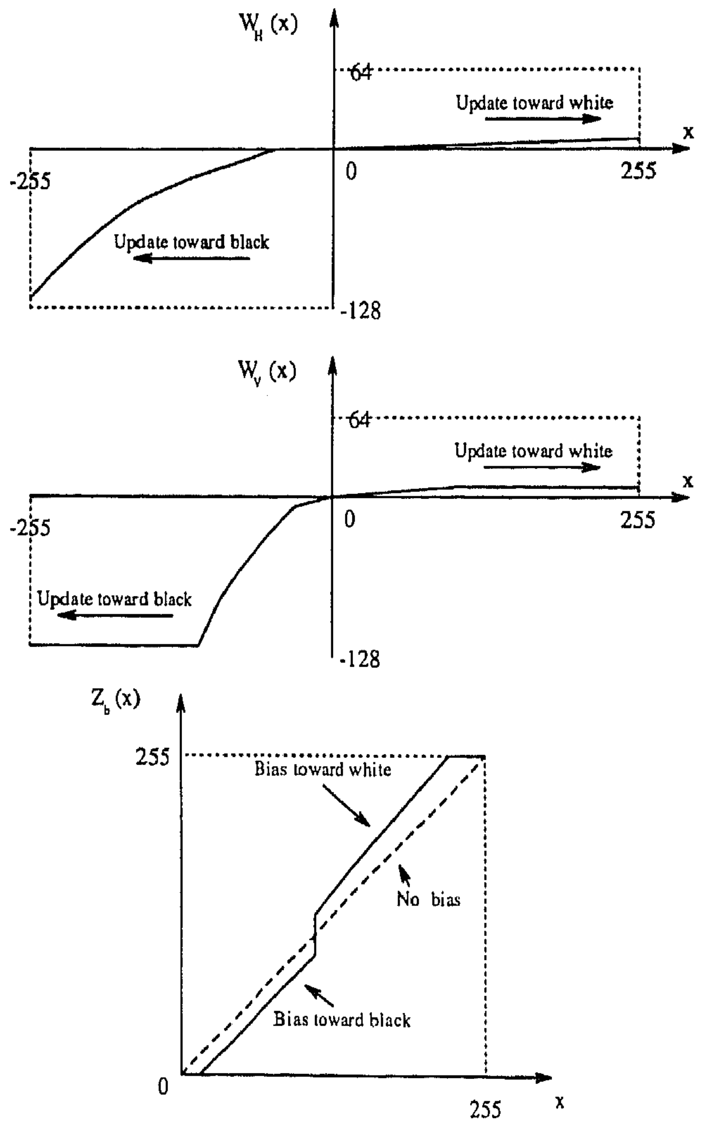


FIG. 3. The update functions $W_H(x)$, $W_V(x)$ and bias function $Z_b(x)$ used in Nonlinear Adaptive Technique.

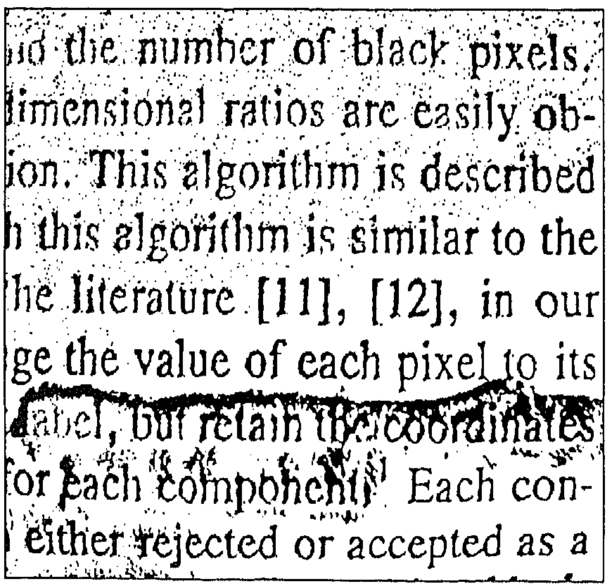

id the number of black pixels.
dimensional ratios are easily ob-
ion. This algorithm is described
h this algorithm is similar to the
he literature [11], [12], in our
ge the value of each pixel to its
label, but retain the coordinates
for each component. Each con-
either rejected or accepted as a

FIG. 5. Binary document character image extracted using Nonlinear Adaptive Technique.

image is also extracted using the same update functions and bias function but different local average which is calculated based on the pixels in the right-bottom direction of the processed pixel. Then the character/graphics image is obtained by performing an *AND* operation on the two binary images. On one hand, almost every character/graphics pixel in one of the two images is also a character/graphics pixel in the other so that it is kept in the final result image, on the other hand, most of the shadow and large dark area edge pixels in one image are not extracted again in the other image so that they are removed in the final result image. However, our experiments has shown that this technique has two disadvantages. One disadvantage is that the strokes are also significantly eroded, the other is its low speed. Although this technique can be extended by including images calculated from right-top and left-bottom directions, it will not be further researched due to its two disadvantages.

2.4. *Integrated Function Technique*

This technique as presented in [8] is virtually equivalent to the second derivative technique evaluated by Palumbo, Swaminathan, and Srihari in [7]. It first detects all the pixels lying near an edge (sharp change in gray level) and labels the pixels on the dark side with + and the pixels on the light side with −. The mathematical description of this operation is as follows:

$$S(x, y) = \begin{cases} 0 & \text{if } A(x, y) < T \\ - & \text{if } A(x, y) \geq T \text{ and } ddxy(x, y) < 0 \\ + & \text{if } A(x, y) \geq T \text{ and } ddxy(x, y) > 0 \end{cases}$$

where

$$dx(x - i, y - j) = |f(x - i, y - j) - f(x - i - 1, y - j)|;$$

$$dy(x - i, y - j) = |f(x - i, y - j) - f(x - i, y - j - 1)|;$$

T is a predetermined parameter;

$$A(x, y) = \sum_{-1 \leq i \leq 1} \sum_{-1 \leq j \leq 1} [dx(x - i, y - j) + dy(x - i, y - j)];$$

$$ddxy(x, y) = f(x + 2, y) + f(x - 2, y) + f(x, y + 2) + f(x, y - 2) - 4 \times f(x, y).$$

Considering all the pixels along with some straight line passing through the currently processed point (x, y), we get a sequences of +, − and 0. If the processed pixel is a character/graphics pixel, then this sequence should be bounded by ordered sequences as illustrated below,

$$-, +, \ldots, [S(x, y) = 0 \text{ or } +], \ldots, +, -$$

and the distance between (−+, +−) is the "stroke width" along this line. Background pixels tend not to be bounded by such ordered sequences. Therefore character/graphics pixels can be extracted out by examing the two sequences corresponding to the horizontal and vertical straight lines passing through the currently processed pixel. If these two sequences are bounded and one of two "stroke widths" is in the predetermined stroke width range, then the processed pixel is a character/graphics pixel; otherwise it is a background pixel.

2.5. *Local Contrast Technique*

In their patent for character recognition applications [11], Giuliano, Paitra, and Stringa presented this technique which is virtually a window operator. Every pixel in the output image is calculated using the following operation on the 3 × 3 × 5 pixels in a window as shown in Figure 6:

begin
  if ($f(x, y) < T_1$) then $b(x, y) = 1$;
  else
    begin
      a_1 = Average of 9 pixels in area A_1;
      $A_{2t} = \{(x, y) \mid (x, y) \text{ in } A_2 \text{ and } f(x, y) > T_2\}$;
      a_2 = Average of pixels in area A_{2t};
      if $(((T_3 \times a_2) + T_5) > (T_4 \times a_1))$ then $b(x, y) = 1$;
      else $b(x, y) = 0$;
    end;
end.

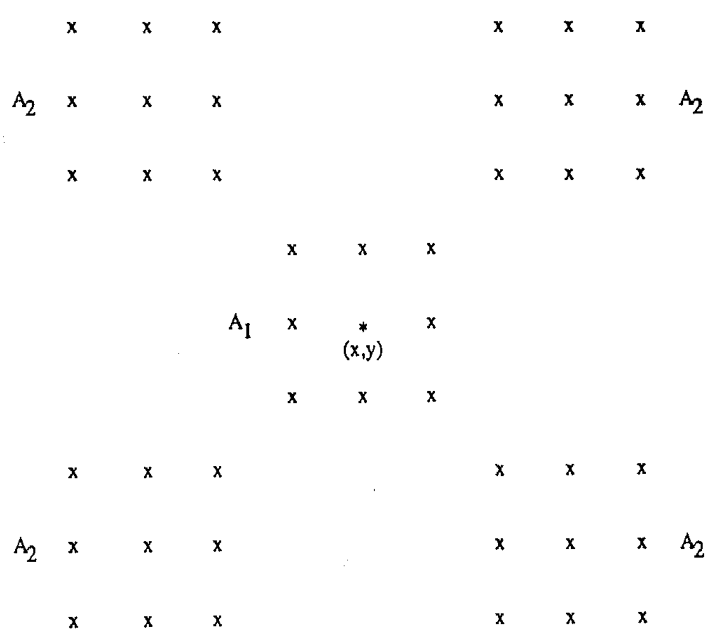


FIG. 6. Window operations in Local Contrast Technique.

where

T_1, a predetermined parameter, is used to extract all the pixels whose gray levels are less than it as character/graphics pixels (similar to the parameter in global thresholding techniques);

T_2, a predetermined parameter, is used to detect all the pixels in A_2 with gray levels not greater than it as unconsidered pixels;

T_3, T_4, T_5, three predetermined parameters, are used for comparing a_1 with a_2.

3. TWO NEW TECHNIQUES

3.1. *Logical Level Technique*

YDH technique, the nonlinear adaptive technique and the local contrast technique have a common disadvantage that they extract the edges of large dark areas. The reason for this is that they have only partially factored the stroke width restriction by a comparison of two values. One value is the gray level of the processed pixel or some local average in a small neighborhood about the processed pixel. The other value is another some local average in a bigger neighborhood (usually the approximate character-size) about the processed pixel. However, unlike the integrated function technique, they are not sensitive to noise because they use local averages.

The integrated function technique has fully factored the stroke width restriction in the sense that it can remove all the large dark areas completely. Similar to the three techniques mentioned above, this technique also uses derivatives, but further, it labels the processed pixel according to the values of its derivatives, detects all the possible character/graphics pixels using the logical bound on the ordered sequences and finally extracts all the character/graphics pixels whose vertical or horizontal "stroke widths" lie in the predetermined stroke width range. The labeling and logical detecting assure that every large dark area is a connected black blob which is removed from the image in the final extraction phase. Therefore the result images from this technique do not have the unwanted edges of large dark areas. However, this technique is very sensitive to noise because of the sensitivity of the derivatives to noise.

From the observations, we get an outline of a better extraction technique. Conceptually, the idea is to compare the gray level of the processed pixel (if the processed image is noise-free) or its smoothed gray level (if the processed image is noisy) with some local averages in the neighborhoods about a few other neighbouring pixels. Thus more than one comparison is made and the comparison results can be considered as "derivatives." Therefore the labeling, detecting and extracting using the "derivatives," the logical bound on the ordered sequences and the stroke width range can be adopted. Since local averages are not sensitive to noise, these "derivatives" should not be sensitive to noise.

Based on this idea, we develop a new technique called logical level technique. This technique predetermines the stroke width range as $[0, W]$. It processes every pixel by simultaneously comparing its gray level or its smoothed gray level with four local averages in the $(2W + 1) \times (2W + 1)$-size neighborhoods centered at the four points P_i, P_i', P_{i+1}, P_{i+1}' shown in Fig. 7. If for certain i, the (smoothed) gray level is at least T levels below all the four local averages, then the processed pixel is extracted as character/graphics pixel. Mathematically, this technique can be described as follows:

$$b(x, y) = \begin{cases} 1 & \text{if } \bigvee_{i=0}^{3} [L(P_i) \wedge L(P_i' \wedge L(P_{i+1}) \wedge L(P_{i+1}')] \\ & \text{is true} \\ 0 & \text{otherwise} \end{cases}$$

where

W is the predetermined maximal stroke width;
$P_i' = P_{(i+4) mod 8}$, for $i = 0, \ldots, 7$;
$L(P) = ave(P) - g(x, y) > T$;
T is a predetermined parameter;
$ave(P) = \sum_{-W \le i \le W} \sum_{-W \le j \le W} f(P_x - i, P_y - j)/(2 \times W + 1)^2$;
P_x, P_y are the coordinates of P;
$g(x, y) = f(x, y)$ or its smoothed value.

In order to reduce the computation, we adopt the following fast algorithms:

1. A fast algorithm to calculate local average

$$a_H(x, y) = \sum_{-W \le i \le W} f(x - i, y) = a_H(x - 1, y) - f(x - W - 1, y) + f(x + W, y)$$

$$a_V(x, y) = \sum_{-W \le j \le W} a_V(x, y - j) = a_V(x, y - 1) - a_H(x, y - W - 1) + a_H(x, y + W)$$

$$ave(x, y) = \frac{a_V(x, y)}{(2 \times W + 1)^2}.$$

2. Logical decomposition:

$$\bigvee_{i=0}^{3} [L(P_i) \wedge L(P_i') \wedge L(P_{i+1}) \wedge L(P_{i+1}')] = \left\{ \bigvee_{i=0,2} [L(P_i) \wedge L(P_i')] \right\} \wedge \left\{ \bigvee_{i=1,3} [L(P_i) \wedge L(P_i')] \right\}.$$

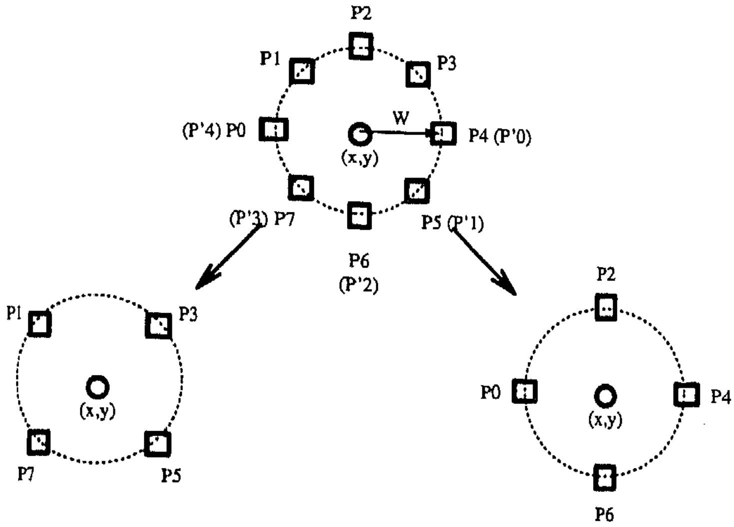


FIG. 7. Decomposition of logical detection.

The major advantages of this technique are its low noise sensitivity, full factoring of the stroke width and high speed. This technique is not sensitive to noise since it uses local averages as basic computing units. Since few of the large dark area pixels satisfy the logical restriction, large dark areas are removed with their edges together. Although usually not all the *AND* and *OR* operations are performed in both the original and fast algorithms of this technique, we can approximate the computation reduction brought by the fast algorithms. The computation reduction in calculating local averages is $[(2W + 1)^2 - 4]$ add/subtraction operations and the reduction in logical comparisons is $(3 - 2)$ *OR* operations and $(12 - 5)$ *AND* operations. Figures 8 and 9 show the result images of this technique.

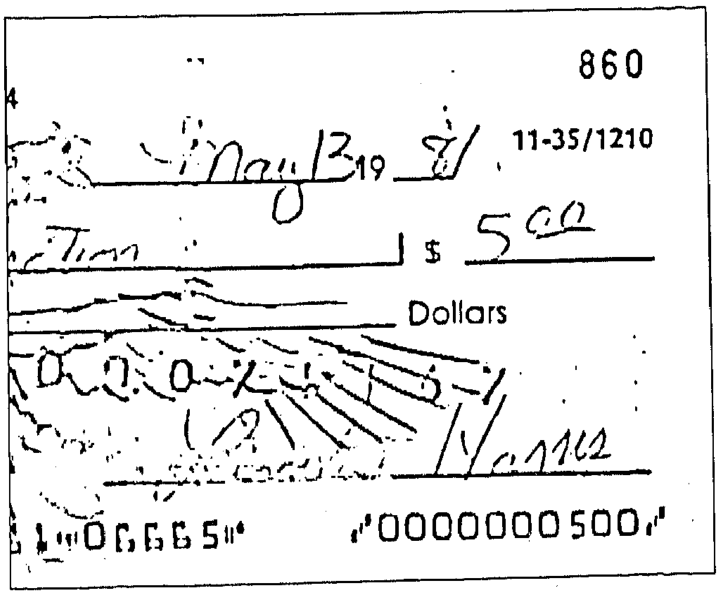


FIG. 8. Binary check character image extracted using Logical Level Technique.

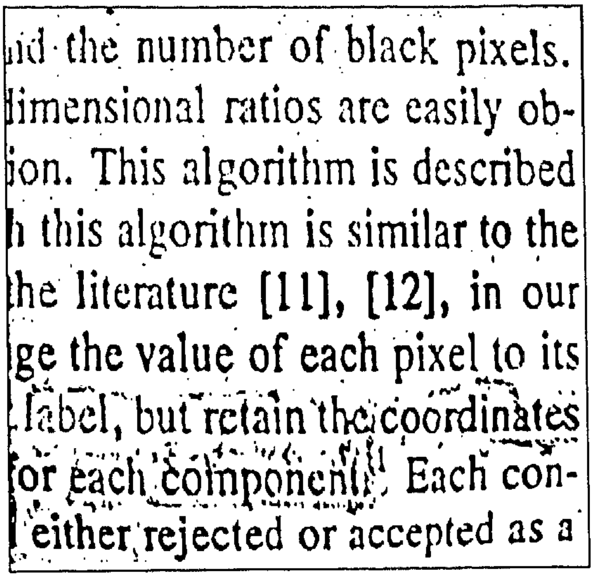

id the number of black pixels.
imensional ratios are easily ob-
ion. This algorithm is described
h this algorithm is similar to the
he literature [11], [12], in our
ge the value of each pixel to its
label, but retain the coordinates
for each component. Each con-
either rejected or accepted as a

FIG. 9. Binary document character image extracted using Logical Level Technique.

3.2. *Mask-Based Subtraction Technique*

This technique predetermines the stroke width range as $[W_{min}, W]$. It considers every original grayscale image as a sum of its background image and its character/graphics image, and consists of four steps. In the first step, most of the background pixels are detected using a logical filter. Considering all the pixels along some straight line passing through the processed pixel, we get a one dimensional signal sequence in which the character/graphics pixels look like some kinds of "particle noise." Therefore we get a logical filter which can be considered as an extension of the median filter for removing particle noise. For more accuracy, this filter is applied to four sequences corresponding to the four straight lines passing through the processed pixel with slopes of 0, $\pi/4$, $\pi/2$ and $3\pi/4$. The resulting image of this step is a binary image in which black pixels are possible character/graphics pixels and white pixels are background pixels. We call this binary image "mask image." In the second step, the mask image is modified by detecting more background pixels using the predetermined stroke width range $[W_{min}, W]$ (refer to Fig. 10). In the third step, for every possible characters/graphics pixel detected out in the modified mask image, its gray level of background image is estimated by a linear interpolation of the four background pixels, $f(P_L)$, $f(P_R)$, $f(P_U)$, $f(P_D)$, in the modified mask image (refer to Fig. 10). In the last step, a grayscale characters/graphics image is obtained by subtracting the estimated background image from the original image, and the binary characters/graphics image is extracted out by applying the global thresholding technique to the grayscale characters/graphics image. The mathematical description of this technique is as follows:

1. Mask image detection using logical filter

```
begin
  if at least W pixels of {f(x + i, y), −W ≤ i ≤ W}
    have values greater than f(x, y),
        then b1(x, y) = 1;
  else
    if at least W pixels of {f(x, y + i), −W ≤ i ≤ W}
      have values greater than f(x, y),
          then b1(x, y) = 1;
    else
      if at least W′ pixels of {f(x + i, y + i), −W′ ≤ i ≤ W′}
        have values greater than f(x, y),
            then b1(x, y) = 1;
      else
        if at least W′ pixels of {f(x − i, y + i), −W′ ≤ i ≤ W′}
          have values greater than f(x, y),
              then b1(x, y) = 1;
        else b1(x, y) = 0;
end.
```

where

W' is the integer closest to $W/\sqrt{2}$;
$b_1(x, y)$ is 1 for a possible character/graphics pixel and 0 for a background pixel.

2. Mask image modification using stroke width range

```
begin
  if b1(x, y) = 1;
    detect dR, dL, dU, dD;
    width = min(dU + dD, dR + dL);
    if width < Wmin or width > W, then b1(x, y) = 0;
end.
```

3. Background image estimation using interpolation

begin
  if $b_1(x, y) = 1$;
    detect d_R, d_L, d_U, d_D;
    if $d_R > W$ or $d_L > W$, then $$f_b(x, y) = \frac{f(x, y - d_U)(1/d_U) + f(x, y + d_D)(1/d_D)}{(1/d_U) + (1/d_D)};$$
    else
      if $d_U > W$ or $d_D > W$, then $$f_b(x, y) = \frac{f(x + d_{R,y})(1/d_R) + f(x - d_{L,y})(1/d_L)}{(1/d_R) + (1/d_L)};$$
      else
        $$f_b(x, y) = \frac{f(x, y - d_U)(1/d_U) + f(x, y + d_D)(1/d_D) + f(x + d_{R,y})(1/d_R) + f(x - d_{L,y})(1/d_L)}{(1/d_U) + (1/d_D) + (1/d_R) + (1/d_L)};$$
  else
    $f_b(x, y) = f(x, y)$;
end.

4. Character/graphics image extraction

$$b(x, y) = \begin{cases} 1 & \text{if } f_b(x, y) - f(x, y) > T \\ 0 & \text{otherwise} \end{cases}$$

where T is a predetermined parameter.

The major advantage of this technique is that it has fully factored the stroke width since it examines every possible character/graphics pixel using the predetermined stroke width range. Figures 11 and 12 show the result images of this technique. This technique has some disadvantages such as low speed, and large memory requirement which will be discussed in the next section.

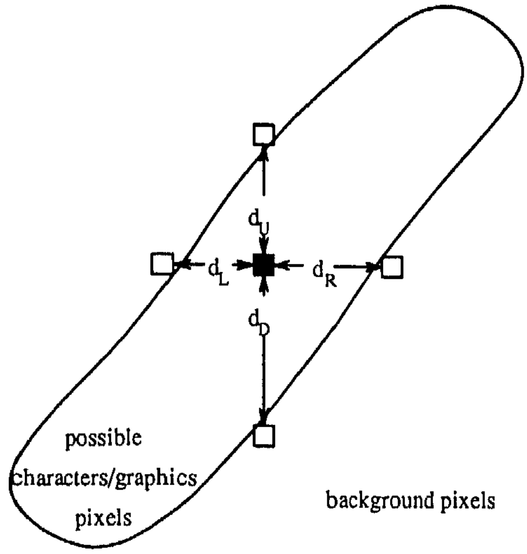


FIG. 10. Four points used in mask image modification and background image estimation.

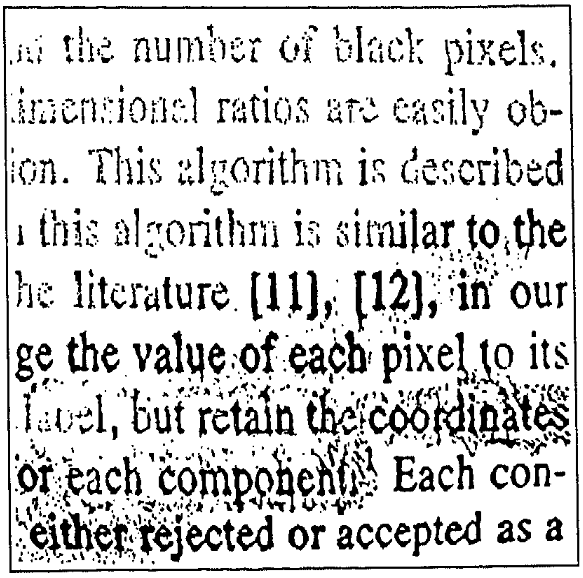

...the number of black pixels.
...imensional ratios are easily ob-
...ion. This algorithm is described
...this algorithm is similar to the
...he literature [11], [12], in our
...ge the value of each pixel to its
...label, but retain the coordinates
of each component. Each con-
either rejected or accepted as a

FIG. 12. Binary document character image extracted using Mask-Based Subtraction Technique.

4. EVALUATION, ANALYSIS, AND EXPERIMENTAL RESULTS

Since there are many evaluation aspects for extraction techniques and there are conflicting requirements for these aspects, it is difficult to find a technique which is superior in every aspect. The lack of quantitative measures for some aspects such as subjective evaluation of output images and parameter setting makes the evaluation more difficult. Therefore the best evaluation is testing in practical applications.

Evaluations of three of the extraction techniques have been reported in [3] and [7]. However, these evaluations did not consider the memory requirement which is very important for low-cost applications.

In this section, we first discuss six evaluation aspects or measures, then systematically evaluate and analyse both new and published techniques with respect to these six aspects based on theoretical analysis and experiments. We implemented and applied the eight techniques to a number of test images. We present here two representative test images shown in Figs. 1 and 2 on which we base our evaluation. One is a typical personal check image with width of 640, height of 512, and gray level range of [0, 255]; the other is a common text document image with highlighted characters and shadows, and with width of 512, height of 480, and gray level range of [0, 255]. All the implementations and experiments were done using software written in C programming language in the UNIX operating system on micro VAX-II computer. Table 1 gives a summary of this evaluation and analysis.

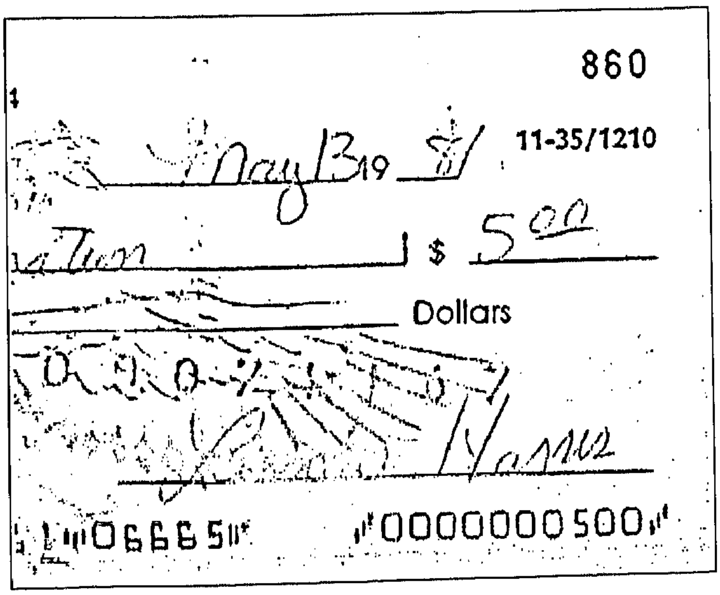


FIG. 11. Binary check character image extracted using Mask-Based Subtraction Technique.

4.1. *Six Evaluation Aspects*

1. *Subjective Evaluation of Result Images.* While some applications require machine recognition and the performances of the extraction technique involved can be measured in terms of recognition ratio, human subjective evaluation is still a reasonable and widely adopted evaluation aspect especially for the applications requiring human judgment. For example, in videoconferencing systems, the transmitted document images are used for manual reading. In check image processing systems, the extracted character image is also sometimes used for manual reading. Even when automated readers are expected to be used, it is still reasonable to use such measure rather than using recognition ratios as the latter may

TABLE 1
Evaluation and Analysis of Seven Character/Graphics Extraction Techniques

Technique	Subjective evaluation	Width*	CPU time	Memory requirement	Parameter number	Parameter setting
Global threshold	Worst	Not	1.1	0	1	Easy
YDH	Block lines, unwanted edges	Partly	74.8	MN	5	Easy
Nonlinear function	Pseudo-shadows unwanted edges	Partly	6.1	M	1276	Difficult
Integrated function	Noise	Fully	14.5	$(2W + 3)M$	2	Easy
Local contrast	Unwanted edges edge	Partly	103.0	$9M$	4	Difficult
Logical level	Best	Fully	7.7	$(4W - 1)M$	2	Easy
Mask-based subtraction	Better than some others	Fully	30.3	$2MN$	3	Easy

Note. Width*, how the stroke width restriction was factored. CPU time, the CPU time (represented in second) used for processing the check image. Memory, the number of registers used to store the intermediate results. M, N, the column number and line number of the processed image. W, the predetermined maximal line width.

depend on the software and/or hardware used for the recognition. In any case it is expected that good visual quality will lead to improved recognition ratios.

2. *Memory*. Since the extraction is often completed along with an optical scanning process, it is often implemented with minimal software and/or hardware within scanners or personal computers. Therefore memory is a key measure for evaluating techniques in low-cost small scale applications. This memory is used to store intermediate results and can be measured in terms of number of registers or bits. In this paper, we adopt the number of registers measure. Therefore an intermediate result data is considered to occupy the same-length register no matter what its data type is.

3. *Speed*. Speed is mostly of concern to the users, especially those who have high volume documents to be processed. Speed is mainly determined by the computation but also depends on the software/hardware implementation. Here we will use CPU time as an inverse proportion of speed.

4. *Stroke Width Restriction*. All extraction techniques except the global thresholding technique have either fully or partly factored this restriction. This restriction comes from the fact that most character/graphics encountered in real application consists of line drawings with almost uniform stroke width. However, for images which contain character/graphics of different stroke widths, factoring this restriction will require adapting the parameters to change from one region to another.

5. *Number of Parameters*. Each of the techniques considered has some parameters that need to be predetermined. For example, the global thresholding technique has one and the local contrast technique has five. Generally, this measure reflects the complexity of using the extraction technique.

6. *Parameter Setting*. Parameter setting is another measure for technique complexity. For every parameter, an initial value needs to be predetermined or set through either some estimation from the processed document or manual direct setting. For example, the only parameter of the global thresholding technique can be set either based on histogram properties [19, 20], "busyness" properties [21, 22], global edge information [23, 24] or by the user. Therefore the parameter setting for the global thresholding is easy and simple. But for the nonlinear adaptive techniques, three functions W_H, W_V, and Z_b need to be predetermined or set for every processed document by the user, and no systematic method or detailed guidance for setting these functions were provided [8]. Therefore the parameter setting for the nonlinear adaptive technique is difficult and complicated. This aspect should be given much consideration when the techniques are to be used by non-professional users.

4.2. *Analysis and Experimental Results*

1. *Global Thresholding Technique*. Global thresholding technique has the highest speed due to its most simple computation, minimal memory requirement due to no intermediate results, smallest parameter number due to only one parameter, and easiest parameter setting due to the availability of various systematic setting methods [19–24]. However, this technique can only handle "ideal" document images of which the gray level distri-

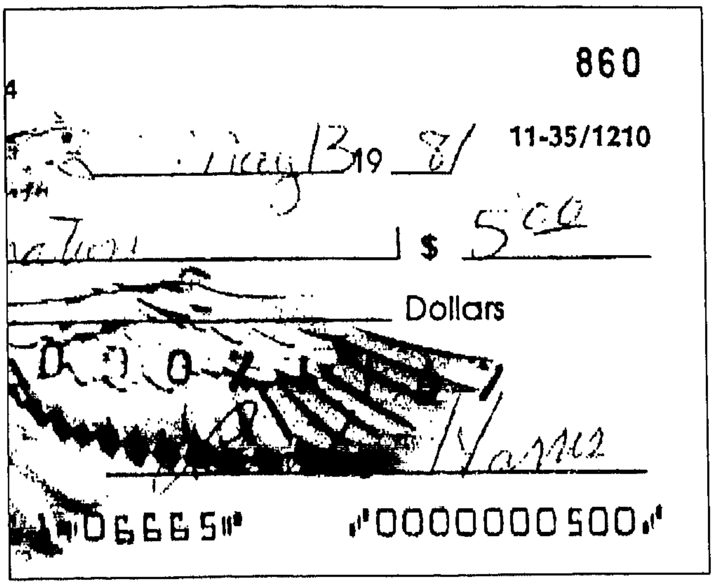


FIG. 13. Binary check character image extracted using Global Thresholding Technique.

butions of character/graphics pxiels and background pixels are well separated away from each other. It can not be applied to the poor-quality documents and checks of which the gray level distributions of character/graphics pixels are often buried within those of background pixels. Figures 13 and 14 give the results of applying this technique to the two test images. For these two result images, the parameter was manually set to the optimal value based on subjective evaluation. It is obvious that many character/graphics pixels have been lost as background pixels and many background pixels have been misextracted as character/graphics pixels. The reason for this is that this technique does not consider the difference between character/graphics pixels and background pixels and processes every pixel based on its own value without consideration of its neighbouring pixels.

2. *YDH Technique*. This technique has partially factored the stroke width restriction through dividing the processed image into blocks of size $T_2 \times T_2$. As pointed out in [18], if T_2 is too large, the background will still contain wide variations within the blocks, so that this technique does not work. On the other hand, if T_2 is too small, a big character could be mistaken to be background and removed from the processed image. However, because of the discontinuity between two adjacent blocks, this dividing may bring a pseudo line at the boundary between two blocks. Another disadvantage is that this technique also extracted the edges of a large dark area in the background such as the eagle's wing in the check image and the highlight area in the text document image. These unwanted edges and pseudo-block lines can be founded clearly in the result images of this technique shown in Figs. 15 and 16. This technique consists of a series of computation on the whole image, therefore its speed is not high and it requires approximately $M \times N$ registers to store its intermediate results. Although the parameter number of this technique is five, the parameter setting is easy since the physical meaning of these five parameters is clear, and T_2, T_4, and T_5 have been suggested by Yasuhiko, Dubois, and Huang to be 16, 128, and 20 respectively.

3. *Nonlinear Adaptive Technique*. The important advantage of this technique is its high speed. It also requires approximately M registers only. However, from its result images shown in Figs. 4 and 5, we can see that some pseudo-shadows were created in the background and that

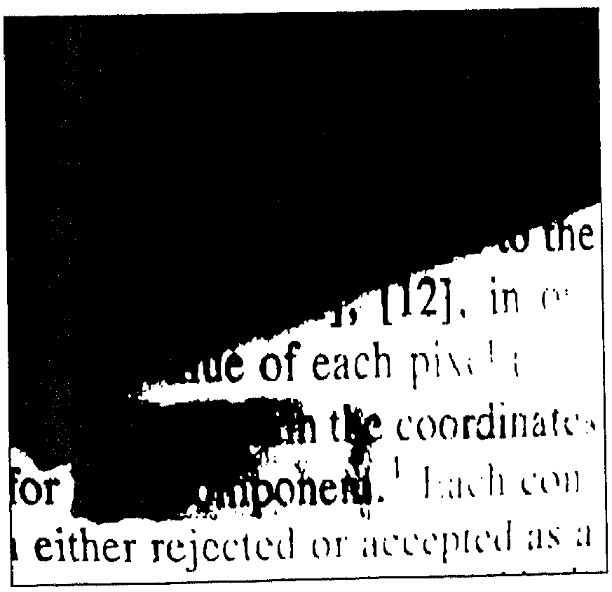


FIG. 14. Binary document character image extracted using Global Thresholding Technique.

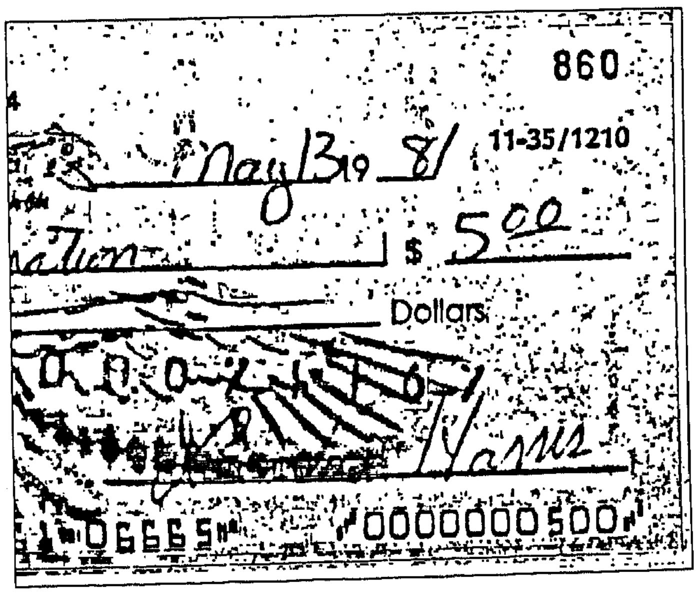
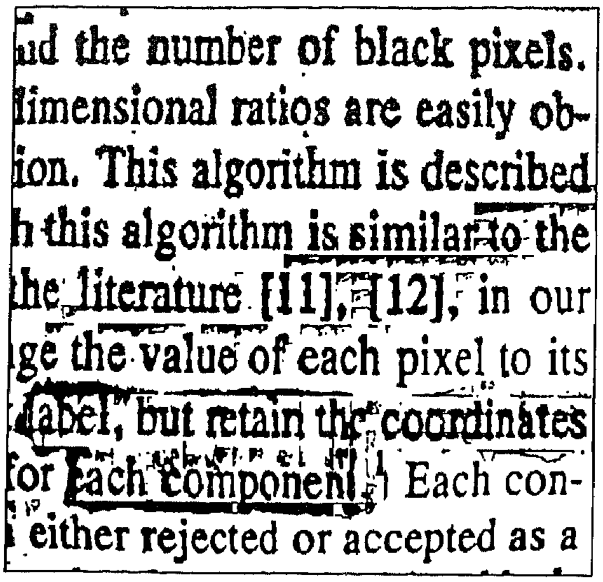


FIG. 15. Binary check character image extracted using YDH Technique.



d the number of black pixels.
imensional ratios are easily ob-
ion. This algorithm is described
h this algorithm is similar to the
he literature [11], [12], in our
ge the value of each pixel to its
label, but retain the coordinates
or each component. Each con-
either rejected or accepted as a

FIG. 16. Binary document character image extracted using YDH Technique.

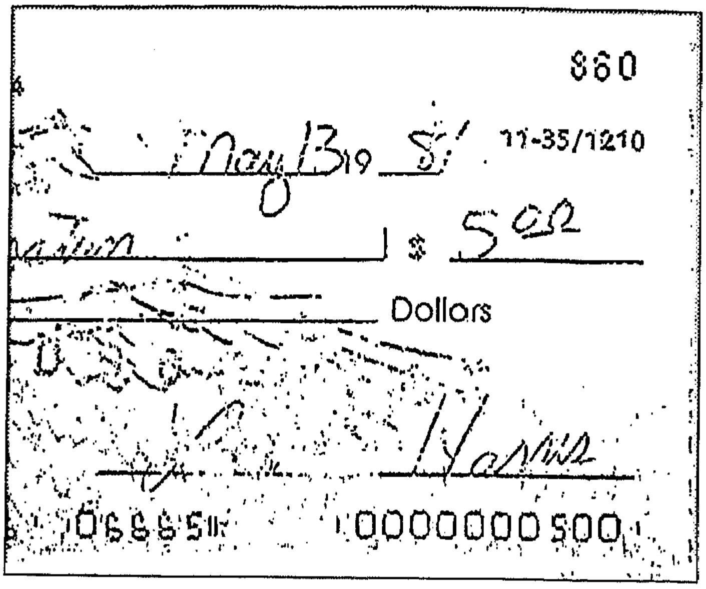
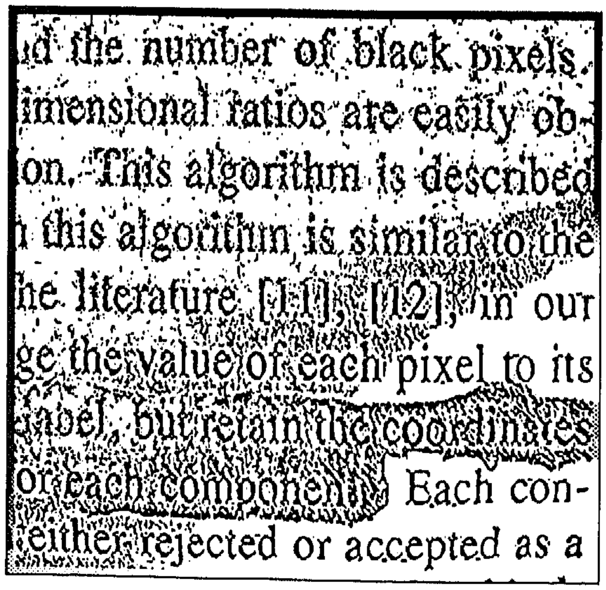


FIG. 17. Binary check character image extracted using Integrated Function Technique.

the left-top part of the dark areas in the background, such as the highlight dark area in the text document image, was also extracted. There are two reasons for this. One is that the local average $f_V(x, y)$ was calculated based on the pixels in the left-top direction of the processed pixel only. The other is that this technique has partially factored the stroke width restriction through its two update functions W_U and W_V. For different images, different values should be set to the two update functions. Although White and Rohrer [8] have given some basic suggestions for setting the update functions, the setting is still very difficult because there are no systematic methods or detailed guidances. For the two test images, although many other settings have been tried, the functions in Fig. 3 seem to give the optimal results shown in Figs. 4 and 5. Although when setting the update or bias functions, people usually choose mathematical functions rather than individual values, we can approximate the number of parameters as follows. Since the argument range of the two update functions is $[-255, 255]$ and $W_H(0)$, $W_V(0)$ have been fixed as zeros, there are 2×510 parameters $W_H(i)$, $W_V(i)$, $1 \leq |i| \leq 255$, to be predetermined. On the other hand, for the bias function, there are 256 predetermined parameters $Z_b(i)$, $0 \leq i \leq 255$. Therefore, the numbers of parameters is approximately 1276.

4. *Integrated Function Technique.* Of the five published techniques, this technique is the only one which fully factors the stroke width restriction. After detecting all the possible character/graphics pixels using first- and second-order derivatives, it extracts only these pixels which are in the bounded sequences and whose vertical or horizontal "stroke widths" lie in the predetermined stroke width range. For the result images of this technique shown in Figs. 17 and 18, the stroke width range was predetermined as $[0, W]$ where W is the maximal stroke width and had value of 4 for the check image and 6 for the text document image. We can see that the major disadvantage of this technique is the noise in its result images. This noise comes from the sensitivity of derivatives used by this technique. However, the speed of this technique is moderate (refer to Table 1), parameter number is only two (T and W), and parameter setting is easy since the physical meanings of T and W are obvious so that the user can easily set values for them according to the processed image. Since only the pixels in the current line and the $(W + 1)$ lines above and below are needed for processing a pixel, the memory requirement of this technique is $(2W + 3)M$.



d the number of black pixels
imensional ratios are easily ob-
on. This algorithm is described
this algorithm is similar to the
he literature [11], [12], in our
ge the value of each pixel to its
abel, but retain the coordinates
or each component. Each con-
either rejected or accepted as a

FIG. 18. Binary document charcter image extracted using Integrated Function Technique.

5. *Local Contrast Technique.* Figures 19 and 20 show the result images of this technique. Compared with the result image of other four published techniques, the images in Figs. 19 and 20 have the best performance with respect to subjective evaluation. The memory requirement is small ($9 \times M$ registers) since this technique is virtually a 9×9 window operation. However, this technique has three disadvantages. The first is its low speed (refer to Table 1). The second is its large number of parameters and difficulty of setting. Since only two out of the T_3, T_4, and T_5 are independent, the number of parameters is 4. T_1, T_2 have easily-understood physical meaning and are easy to set. For a given image, it is not clear how to set the values of T_3, T_4, and T_5. Palumbo, Swaminathan, and Srihari [7] set T_1, T_2, T_3, T_4, and T_5 as 20, 20, 0.85, 1, and 0 for their test images. But our result images using this parameter setting look very unsuccessful. After many trials, we got a setting (40, 40, 1, 1, −3) whose result images look best, shown in Figs. 19 and 20. Different parameter settings can yield quite different results for a given image. The third disadvantage is that this technique also extracts the edges of large dark areas in the background, such as the highlight dark area in the text document image.

6. *Logical Level Technique.* From the result images shown in Figs. 8 and 9, we can see that this technique has the best performance in terms of subjective evaluation. On one hand, its result images do not look as noisy as those of the integrated function technique. On the other hand, unlike the result images of YDH technique, the nonlinear adaptive technique and the local contrast technique, its result images do not contain the unwanted edges of large dark areas in the background, therefore this technique has fully factored the stroke width restriction. Compared with the integrated function technique which has also factored the stroke width restriction, this technique has the following similarities. Every logical comparison $L(P)$ is similar to the comparison $A(x, y) > T$; every $L(P_i) \wedge L(P_i')$ is similar to the bound on the ordered sequences along the line passing through P_i and P_i'. This technique implicitly assumes the predetermined stroke width range is $[0, W]$. In order to make more accurate extraction, four points P_i, P_i', P_{i+1} and P_{i+1}' are used instead of the two points P_i and P_i'. This technique has only two parameters T and W and the parameter setting is very easy because both T and W have obvious physical meaning. The memory requirement is $(4W - 1)M$ registers which can be met by usual low-cost hardware because W is usually small. In addition, this technique has very high speed which is almost equal to that of the nonlinear adaptive technique (refer to Table 1).

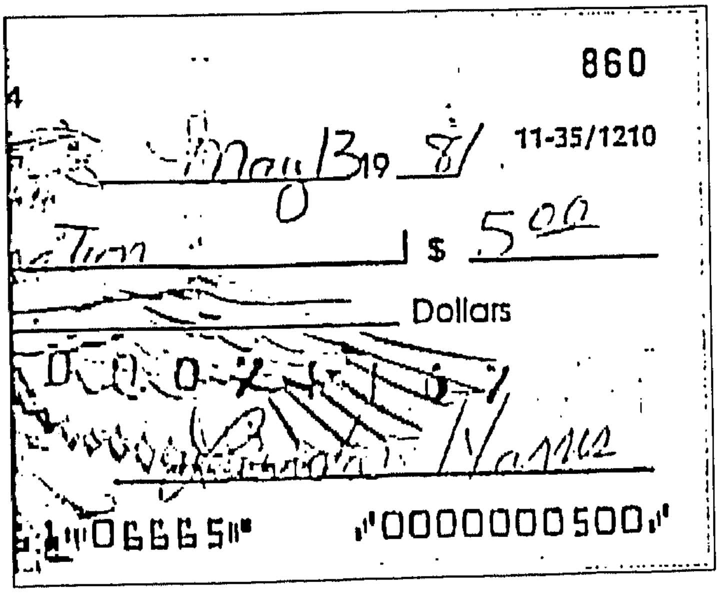


FIG. 19. Binary check character image extracted using Local Contrast Technique with ($T_1 = 50$, $T_2 = 50$, $T_3 = 1$, $T_4 = 1$, $T_5 = -15$).

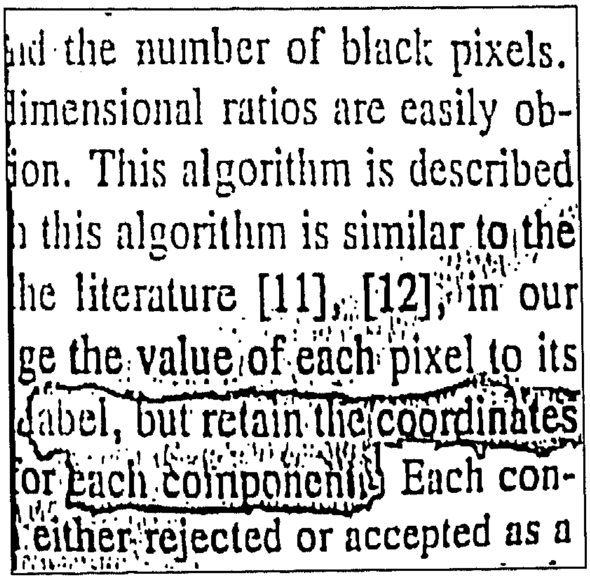

nd the number of black pixels.
imensional ratios are easily ob-
ion. This algorithm is described
n this algorithm is similar to the
he literature [11], [12]; in our
ge the value of each pixel to its
label, but retain the coordinates
or each component. Each con-
either rejected or accepted as a

FIG. 20. Binary document character image extracted using Local Contrast Technique with ($T_1 = 40$, $T_2 = 40$, $T_3 = 1$, $T_4 = 1$, $T_5 = -3$).

7. *Mask-Based Subtraction Technique.* Unlike the other two new techniques which are based on the previous works, this technique comes from an independent idea. Figures 11 and 12 show the result images of this technique. Figures 21 and 22 show the modified mask images. Compared with the result images of the five published techniques, the result images of this technique look better than those of YDH technique, integrated function technique and nonlinear adaptive technique, but worse than those of local contrast technique. It has only three parameters W_{min}, W, and T whose parameter setting is easy. It has fully factored the stroke width restriction because it examines every possible character/graphics pixel using the predetermined stroke width range. However this technique seems inappropriate to handle mixed text font and size documents, which is revealed by the result images. This technique has another two disad-

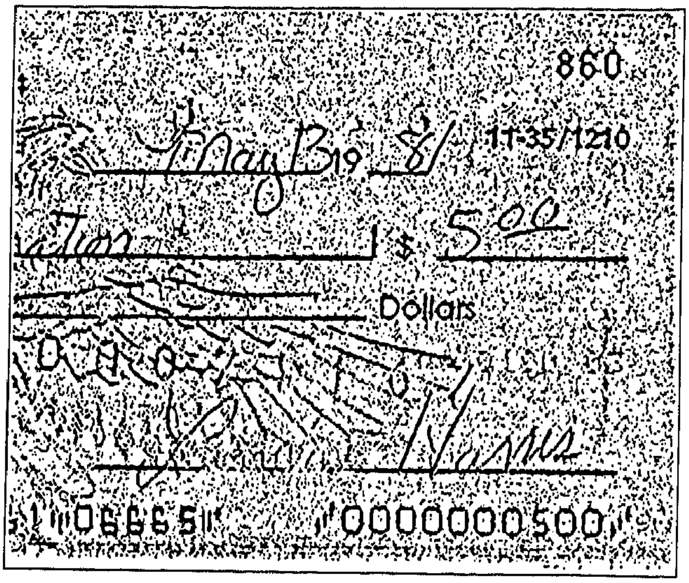


FIG. 21. Modified binary check mask image obtained using Mask-Based Subtraction Technique.

vantages. One is that it requires approximately $2 \times M \times N$ registers of which $M \times N$ registers to store the binary image b_1 and other $M \times N$ registers to store the original image. The other disadvantage is its low speed (refer to Table 1). These two disadvantages restrain its use for high-speed low-cost applications.

5. SUMMARY AND CONCLUSIONS

Aiming at high-speed low-cost applications, this paper reviews five published extraction techniques and presents two new techniques, of which one is an improvement on previous work and the other is independent. Systematical evaluation and analysis of these eight techniques with experiments on images of a typical check and a poor-quality text document have shown that one of the new techniques, the logical level technique, is superior to the other seven with respect to subjective evaluation. The analysis has also shown that this technique is most suitable for high-speed low-cost applications due to its high speed, small parameter number, easy parameter setting, and minimal memory requirement.

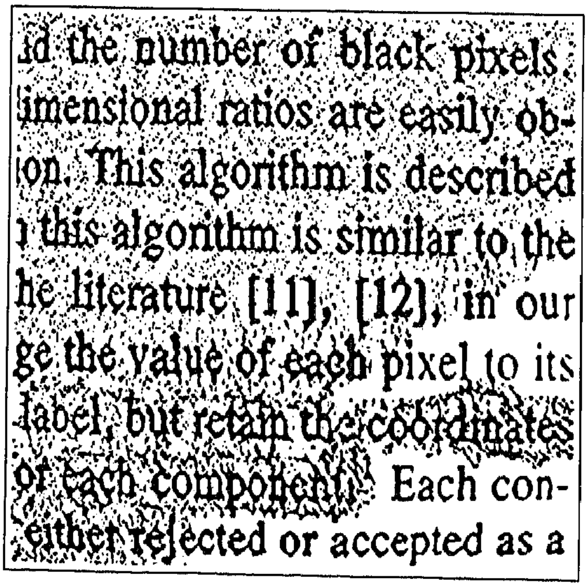

id the number of black pixels
imensional ratios are easily ob-
ion. This algorithm is described
i this algorithm is similar to the
he literature [11], [12], in our
ge the value of each pixel to its
label, but retain the coordinates
of each component. Each con-
either rejected or accepted as a

FIG. 22. Modified binary document mask image obtained using Mask-Based Subtraction Technique.

REFERENCES

1. Calera Recognition Systems, Inc., *TrueScan: Software Developer's Manual.*
2. V. Nagasamy and N. A. Langrana, Engineering drawing processing and vectorization system, *Comput. Vision Graphics Image Process.* **49**, 1990, 379–397.
3. H. H. Chang, Image thresholding for blueprint image extraction, *Conference Proceedings: Vision '87*, Detroit, Michigan, June 8–11, 1987, pp. 5-1–5-18.
4. R. Kasturi and J. Alemany, Information extraction from images of paper-based maps, *IEEE Trans. Software Eng.* **14**(5), May 1988, 671–675.
5. A. C. Downton and C. G. Leedham, Preprocessing and presorting of envelope images for automatic sorting using OCR, *Pattern Recognition* **23**(3/4), 1988, 347–362.
6. S. N. Sirhari, Feature extraction for locating address blocks on mail pieces, *From Pixels to Features*, pp. 261–273, Elsevier, Amsterdam, 1989.
7. P. W. Palumbo, P. Swaminathan, and S. N. Srihari, Document image binarization: evaluation of algorithms, in *Proceedings, SPIE Symposium on Applications of Digital Image Processing IX*, 1986, Vol. 697, pp. 278–285.
8. 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), July 1983, 400–411.
9. J. L. Mitchell, W. B. Pennebaker, D. Anastassiou, and K. S. Pennington, Graphics image coding for freeze-frame videoconferencing, *IEEE Trans. Commun.* **37**(5), May 1989, 515–522.
10. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd ed., Addison–Wesley, Reading, MA, 1987.
11. E. Giuliano, O. Paitra, and L. Stringa, Electronic character reading system, U. S. Patent 4,047,15 (Sept. 6, 1977).
12. L. A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images, *IEEE Trans. pattern Anal. Mach. Intell.* **10**(6), Nov. 1988, 910–918.
13. D. Wang and S. N. Srihari, Classification of newspaper image blocks using texture analysis, *Comput. Vision Graphics Image Process.* **47**, 1989, 327–352.
14. D. G. Elliman and I. T. Lancaster, A review of segmentation and contextual analysis techniques for text recognition, *Pattern Recognition* **23**(3/4), 1990, 337–346.
15. W. Scherl, F. Wahl, and H. Fuchsberger, Automatic separation of text, graphics and picture segments in printed material, in *Pattern Recognition in Practice* (E. S. Gelsema and L. N. Kanal, Eds.), pp. 213–221, North-Holland, Amsterdam, 1980.
16. F. M. Wahl, K. Y. Wong, and R. G. Casey, Block segmentation and text extraction in mixed text/image documents, *Comput. Graphics Image Process.* **20**, 1982, 375–390.

17. M. Nadler, Survery: Document segmentation and coding techniques, *Comput. Vision Graphics Image Process.* **28,** 1984, 240–262.

18. Y. Yasuda, M. Dubois, and T. S. Huang, Data compression for check processing machines, *Proc. IEEE* **68**(7), July 1980, 874–885.

19. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybernet.* **9,** Jan. 1979, 62–66.

20. J. S. Weszka and A. Rosenfeld, Histogram modification for threshold selection, *IEEE Trans. Syst. Man Cybernet.* **9,** Jan. 1979, 38–52.

21. F. Deravi and S. K. Pal, Gray level thresholding using second-order statistics, *Pattern Recognition Lett.* **1,** 1983, 417–422.

22. J. S. Weszka and A. Rosenfeld, Thresholding evaluation techniques, *IEEE Trans. Syst. Man Cybernet.* **8,** Aug. 1978, 622–629.

23. J. Kittler, J. Lllingworth, J. Foglein, and K. Paler, An automatic thresholding algorithm and its performance, *Proc. 7th International Conference on Pattern Recognition,* Montreal, 1984, pp. 287–289.

24. J. Kittler, J. Lllingworth, J. Foglein, and K. Paler, Threshold selection based on a simple image statistics, *Comput. Vision Graphics Image Process.* **30,** 1985, 125–147.