always $M_{pq} \geq m_{pq}$; the error $M_{pq} - m_{pq}$ depends on the given p and q. Some examples are as follows:

$$M_{20} - m_{20} = \frac{m_{00}}{12}$$
$$M_{31} - m_{31} = \frac{m_{11}}{4}$$
$$M_{40} - m_{40} = \frac{m_{20}}{2} + \frac{m_{00}}{80}.$$

Spiliotis and Mertzios analyzed the computing complexity of moment calculation by means of (2) and (3), respectively. To compute all moments of order up to $(L-1, L-1)$ using (2), one needs $2LN$ power calculations, L^2 multiplications and $2LN$ additions. The usage of (3) is more effective for large blocks because its complexity does not depend on the block size. It requires $4L$ power calculations, $2L^2 - L$ multiplications and $L^2 - L$ additions.

When using our refined method (5), the terms

$$\frac{(N+0.5)^{p+1} - 0.5^{p+1}}{p+1}$$

can be precalculated in advance for $p = 0, 1, \ldots, L-1$. Thanks to this, the method requires only $2L$ power calculations, $L^2/2 + L$ multiplications and $2L$ additions, or, equivalently, $L^2/2 + 3L$ multiplications and $2L$ additions. Clearly, this is more effective than the original method.

The difference between the both methods is even more marked when one wants to calculate only one moment of order (p, q). In such a case, (2) requires $2N$ powers and $2N$ additions. The use of the recursive formula (3) is not a good choice: To find S_N^p, we have to evaluate all lower-order sums $S_N^1, \ldots, S_N^{p-1}$, that takes $O(p^2)$ operations. On the other hand, the complexity of moment calculation by means of (5) does not depend on the moment order and the block size either. M_{pq} is obtained after ten elementary operations only.

V. Conclusion

We have presented a refinement of the recently published method for moment calculation based on image block representation. The new technique has been shown to outperform the original one, both in accuracy and computational complexity.

References

[1] I. M. Spiliotis and B. G. Mertzios, "Real-time computation of two-dimensional moments on binary images using image block representation," *IEEE Trans. Image Processing*, vol. 7, pp. 1609–1615, Nov. 1998.

[2] L. Yang and F. Albregtsen, "Fast and exact computation of cartesian geometric moments using discrete Green's theorem," *Pattern Recognit.*, vol. 29, no. 7, pp. 1061–1073, 1996.

[3] M. F. Zakaria, L. J. Vroomen, P. Zsombor-Murray, and J. M. van Kessel, "Fast algorithm for the computation of moment invariants," *Pattern Recognit.*, vol. 20, no. 6, pp. 639–643, 1987.

[4] M. Dai, P. Baylou, and M. Najim, "An efficient algorithm for computation of shape moments from run-length codes or chain codes," *Pattern Recognit.*, vol. 25, no. 10, pp. 1119–1128, 1992.

[5] B. C. Li, "A new computation of geometric moments," *Pattern Recognit.*, vol. 26, no. 1, pp. 109–113, 1993.

[6] J. Flusser, "Fast calculation of geometric moments of binary images," in *Proc. 22nd OAGM'98 Workshop Pattern Recognition Medical Computer Vision*, Illmitz, Austria, 1998, pp. 265–274.

[7] B. C. Li and J. Shen, "Fast computation of moment invariants," *Pattern Recognit.*, vol. 24, no. 8, pp. 807–813, 1991.

[8] X. Y. Jiang and H. Bunke, "Simple and fast computation of moments," *Pattern Recognit.*, vol. 24, no. 8, pp. 801–806, 1991.

[9] W. Philips, "A new fast algorithm for moment computation," *Pattern Recognit.*, vol. 26, no. 11, pp. 1619–1621, 1993.

Morphological Text Extraction from Images

Yassin M. Y. Hasan and Lina J. Karam

***Abstract*—This paper presents a morphological technique for text extraction from images. The proposed morphological technique is insensitive to noise, skew and text orientation. It is also free from artifacts that are usually introduced by both fixed/optimal global thresholding and fixed-size block-based local thresholding. Examples are presented to illustrate the performance of the proposed method.**

***Index Terms*—Labeling, mathematical morphology.**

I. Introduction

Text extraction from images and video sequences finds many useful applications in document processing [1], detection of vehicle license plate, analysis of technical papers with tables, maps, charts, and electric circuits [2], parts identification in industrial automation [3], and content-based image/video retrieval from image/video databases [4], [5]. Educational and training video and TV programs such as news contain mixed text-picture-graphics regions. Region classification is helpful in object-based compression, manipulation and accessibility. Also, text regions may carry useful information about the visual content.

Text extraction may employ binarization [1], [5]–[7] or directly process the original image [4], [8], [9]. In [5], a survey of existing techniques for page layout analysis is presented. In addition, a computationally efficient page decomposition algorithm is introduced for layout analysis of technical journal pages. In such applications, characters are black connected components with high contrast in a homogeneous background. So, simple fixed global thresholding is employed. The algorithm classifies the image regions as text, table, image, or drawing. The main disadvantage of the algorithm of [5] is that small image regions and drawings may be identified as text with large characters. In [1], the original image is globally thresholded. Then, a run-length smearing procedure is employed to group horizontally and vertically close connected components into regions. Finally, knowledge-based schemes are employed to identify the text regions. However, noisy and distorted binary images may result from global thresholding, especially due to varying illumination conditions, reflections, and the effect of the scanner point spread function. In [6], local thresholding has been applied by dividing the original image into blocks with specified size, determining an optimal threshold level per block to be assigned to its center pixel, and finding pixel-wise threshold levels by interpolation. The performance of such scheme critically depends on how efficiently the block size is chosen. Also, interpolating the threshold levels of a high frequency block surrounded by uniform blocks degrades the output binary image. In [7], the constrained runlength algorithm of [10] and connected component labeling are applied to a binary version of the input image in order to divide the image into text and graphics subregions. The histograms of the subregions are then used to classify them as text or graphics. A connected component-based method is proposed in [9];

Manuscript received December 9, 1998; revised May 26, 2000. This work was supported in part by Arizona State University and the Cultural and Educational Bureau, Embassy of Egypt. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Scott T. Acton.

The authors are with the Telecommunications Research Center, Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-7206 USA (e-mail: ymy@asu.edu; karam@asu.edu).

Publisher Item Identifier S 1057-7149(00)09399-4.

this method uses several heuristics in relation to size, alignment, and proximity, in order to extract the text components.

Mathematical morphology is a topological- and geometrical-based approach for image analysis [11]. It provides powerful tools for extracting geometrical structures and representing shapes in many applications. Morphological feature extraction techniques have been efficiently applied to character recognition and document analysis [12], especially if dedicated hardware is used. In this paper, we propose an algorithm for text string extraction based on morphological operations. The proposed algorithm is insensitive to skew and text orientation, is free from artifacts that are introduced by both global and fixed-size block-based local thresholding, and robust to noise. The proposed technique has been successfully tested on many color and gray-scale images and videos.

The paper is organized as follows. In Section II, the proposed morphological text extraction technique is described. Examples and comparison with existing text extraction algorithms are presented in Section III. A conclusion is given in Section IV.

II. Morphological Text Extraction

The proposed algorithm assumes some practical contrast, color, and size restrictions as follows.

1) A text string represents foreground pixels (mostly having the same intensity and width) in an almost homogeneous background with sufficient difference in brightness (a difference $\geq$30 gray levels is assumed in the proposed algorithm) [4]. Hence, text pixels create edges.
2) Each character has a moderate size $L \times W$ (pixels) for its enclosing box and thickness T (pixels) relative to the scanning resolution R (pixels/mm) of the image. Given the input image resolution, the algorithm parameters change automatically to accommodate the allowable minimum and maximum character size and thickness. Based on [13, Ch. 1], the minimum and maximum allowable character size and thickness are determined using the image resolution R as follows:

$$1.8R \leq L \leq 8.4R, \quad 1.5R \leq W \leq 7R, \quad 0.2R \leq T \leq 2R \quad (1)$$

where $R \geq 9$ pixels/mm. Smaller characters are not readable and can not be identified using an OCR system. Larger components are considered as graphics. Also, close edges (within $1.2R$ distance in pixels) in the text regions are guaranteed. Hence, morphological dilation followed by erosion using structuring elements with different sizes, groups the character components into blocks.
3) A text string contains more than one word and each word is at least three characters. So, a large region with small number of connected components is not a text region.

A flow diagram of the proposed algorithm is shown in Fig. 1, and the algorithm steps are summarized as follows.

Step 1) The RGB components of the input color image are combined to give an intensity image Y as follows:

$$Y = 0.299R + 0.587G + 0.114B \quad (2)$$

where R, G and B are the red, green and blue components of the input color image, respectively. The color components may differ in a text region (and/or its surrounding background) while having an almost constant intensity. So, the intensity image Y is processed in the next steps of the algorithm rather than the color components R, G and B. If the input image is originally a gray-level image, the image is processed directly starting at Step 2.

Step 2) An efficient morphological edge detection scheme is applied to the intensity image Y as follows [14]. The image

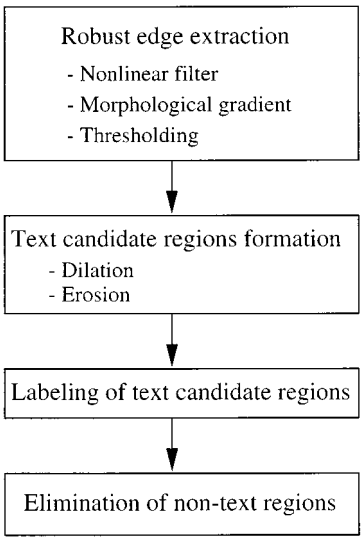

Fig. 1. Flow diagram of the proposed text extraction algorithm.

Y is blurred (to reduce false edges and over-segmentation) using open-close and close-open filters [11]. The final blurred image $Y_{\rm bl}$ is the average of the outputs of these filters, i.e.

$$Y_{\rm bl} = \frac{(Y_B)^B + (Y^B)_B}{2} \quad (3)$$

where B is the 3×3 8-connected structuring element, and Y_B and Y^B denote the opening and closing of Y by the structuring element B, respectively.

Next, the Morphological gradient (MG) operator is applied to the blurred image $Y_{\rm bl}$ resulting in an image e_s as follows:

$$e_s = \mathrm{MG}(Y_{\rm bl}, B) = \delta_B(Y_{\rm bl}) - \varepsilon_B(Y_{\rm bl}) \quad (4)$$

where δ and ε are the dilation and erosion operators, respectively, and B is the 3×3 eight-connected structuring element. The MG is an edge-strength extraction operator that gives symmetric edges between the foreground and background regions.

The resulting image e_s is then thresholded to obtain a binary edge image. A global nonhistogram-based thresholding technique has been incorporated rather than local (adaptive) thresholding in this step [15]. The threshold level γ is determined by

$$\gamma = \frac{\sum(e_s \cdot s)}{\sum s} \quad (5)$$

where "$\cdot$" denotes pixel-wise multiplication and s is given by

$$s = \max(|g_1 * * e_s|, |g_2 * * e_s|). \quad (6)$$

In (6), $g_1 = [-1\ 0\ 1]$, $g_2 = [-1\ 0\ 1]^t$, and "$**$" denotes two-dimensional linear convolution.

The binary edge image (e) is then given by

$$e = \begin{cases} 1, & \text{if } e_s > \gamma \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Step 3) Close edges (within $1.2R$ distance in pixels, where R is the image resolution in pixels/mm) in the binary edge image e are grouped by dilation using eight-connected structuring elements. Then, small connected components in the dilated

```
Definitions:  A: i/p NxM binary image ,   B: o/p NxM labeled image
Algorithm:
    begin
        % Assign temporary labels to nonzero pixels of the image.
        for i=1 to N, j=1 to M, do
           if A(i,j) =1, then sequentially assign unique integer number to B(i,j).
        end do
        set max_temp_label= the maximum assigned integer label.

        % Propagate maximum label per connected component.
        do while at least one pixel value of B(i,j) changes
           % back propagation row-wise
           for i=N to 2, j=1 to M, do
              if both B(i,j) and B(i-1,j) > 0, then
                 set both B(i,j) and B(i-1,j) to maximum(B(i,j),B(i-1,j))
           end do
           % back propagation column-wise
           for i=1 to N, j=M to 2, do
              if both B(i,j) and B(i,j-1) > 0, then
                 set both B(i,j) and B(i,j-1) to maximum(B(i,j),B(i,j-1)
           end do
           % forward propagation row-wise
           for i=1 to N-1, j=1 to M, do
              if both B(i,j) and B(i+1,j) > 0, then
                 set both B(i,j) and B(i+1,j) to maximum(B(i,j),B(i+1,j)
           end do
           % forward propagation column-wise
           for i=1 to N, j=1 to M-1, do
              if both B(i,j) and B(i,j+1) > 0, then
                 set both B(i,j) and B(i,j+1) to maximum(B(i,j),B(i,j+1))
           end do
        end while

        % Sequential relabeling
        for k=1 to max_temp_label, do
           initialize array_label(k)=0
        end do
        max_label=0
        for i=1 to N, j=1 to M, do
           if A(i,j) =1, then
              if array_label(B(i,j)) = 0, then
                 increment max-label
                 array_label(B(i,j))=max_label
                 B(i,j)=max_label
              else
                 B(i,j)=array_label(B(i,j))
        end do
```

Fig. 2. Pseudocode of the proposed labeling algorithm.

image are filtered using erosion. The output is a binary image e_c that contains text candidate regions.

Step 4) Each text candidate region is uniquely labeled by assigning the same unique number for all connected pixels. The proposed labeling algorithm is a fast forward-backward propagation algorithm. The steps of the labeling algorithm are given in Fig. 2 in the form of a pseudocode.

Step 5) Region analysis: For each labeled text candidate region, we find the corresponding region in the intensity image Y. Each gray-level region in Y, corresponding to a labeled text candidate region, is thresholded as in (5) and (6). In this case, "e_s" is replaced by the considered text candidate gray-level region. This adaptive thresholding technique is less sensitive to illumination conditions and reflections than existing methods that are based on global fixed or optimal thresholding [1], [5], and on local thresholding [8].

A thresholded text candidate region may contain some too large and/or too small connected components that do not satisfy (1). Such noncharacter connected components are filtered out of the text candidate re-

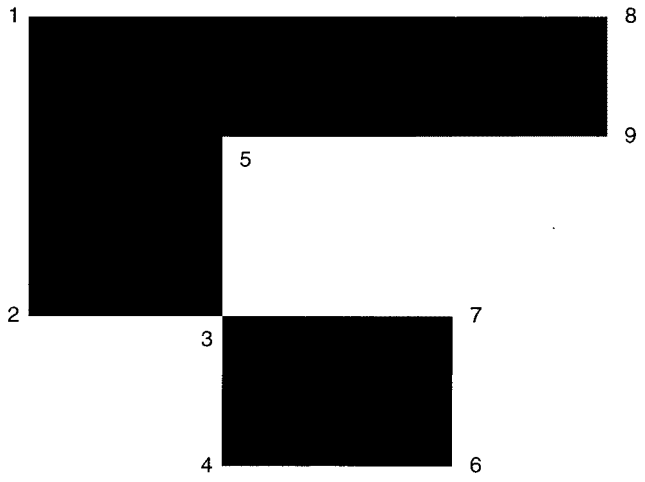

Fig. 3. Types of corners. Outer corners: 1, 2, 4, 6, 7, 8, and 9. Inner corner: 5. Inner-outer corner: 3.

Fig. 4. Text extraction from an advertising display image. (a) The intensity image Y of the original image with text and (b) the extracted text block marked in the image.

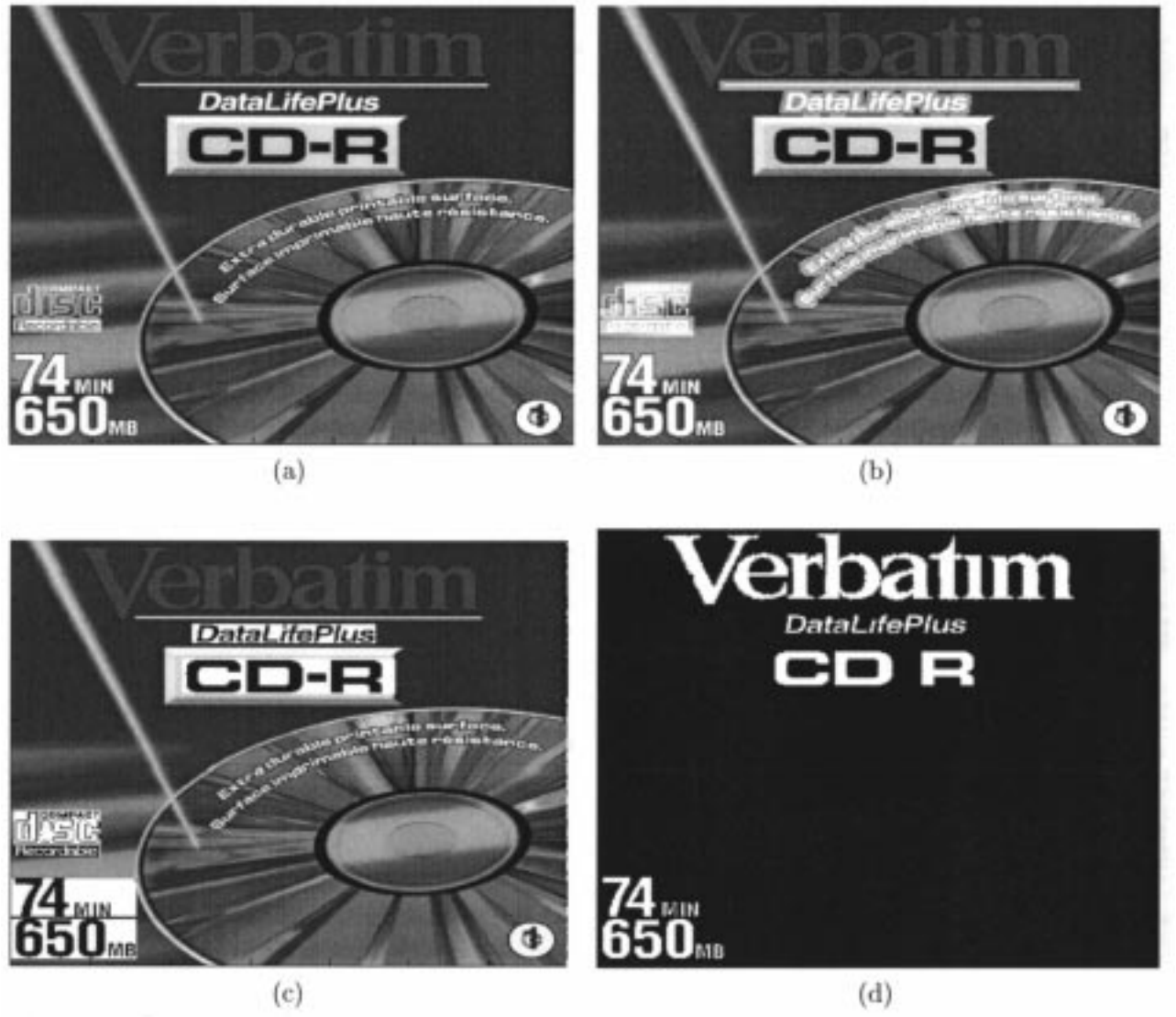

Fig. 5. Text extraction from a CD-cover image. (a) The intensity image Y of the original CD-cover image with text. (b) Results of the proposed algorithm; the extracted text block marked with a white background in the CD-cover image. (c) Results of the algorithm of [7]; the extracted text strings are black with a white background. (d) Results of the algorithm of [9].

gions. In addition, a size-ratio filter is used as in [13] to filter out non-character connected components that satisfy

$$\left(A/(W \times L) \leq \frac{1}{4.5} \text{ or } > .95\right) \quad \text{and} \quad \left(\min(W/L, L/W) \leq \frac{1}{5}\right) \tag{8}$$

where A is the area of the connected component in pixels.

Then, the number of connected components, in each thresholded region, is computed by filling each region using the fast filling algorithm of [16] and applying the hit-or-miss method for counting the number of inner, outer, and inner-outer corners (see Fig. 3) [17], [18]. The number of connected components N_{cc} is then given by [18]

$$N_{cc} = (c_i - c_o + 2c_{i\text{-}o})/4 \tag{9}$$

where c_i, c_o, and $c_{i\text{-}o}$ are the number of inner, outer, and inner-outer corners, respectively. Any region with a number of components less than six (two words) is discarded.

Three successive erosion operations are carried out for each of the remaining regions. Text regions should loose most of their pixels, at least 1/4 of the pixels. Gray-level homogeneity is tested where text should have uniform level. Moreover, the average gray levels of the connected components in each text candidate region are examined to be at least 30 levels different from the average gray level of their surroundings.

More robust text segmentation can be obtained at the expense of more computations by applying successive opening operations to each remaining text candidate region, with structuring element of increasing size, while calculating the number of the pixels with value 1, i.e., calculating the opening-distribution [19]. Then, the normalized opening distribution (open-spectrum) is determined. Out of the remaining regions, a region which has a normalized open-spectrum with a value more than or equal to 0.5 is considered as a text string region. The parameter 0.5 was experimentally determined.

The algorithm stops when all labeled text candidate regions are analyzed as discussed in Step 5.

III. Examples and Discussion

In this section, the performance of the proposed algorithm is illustrated and compared to existing text extraction algorithms, the connected component-based method of [9] and the histogram-based of [7]. Fig. 4(a) contains 90° rotated text and Fig. 5(a) contains a curved text string. Applying the proposed algorithm, Figs. 4(b) and 5(b) show that

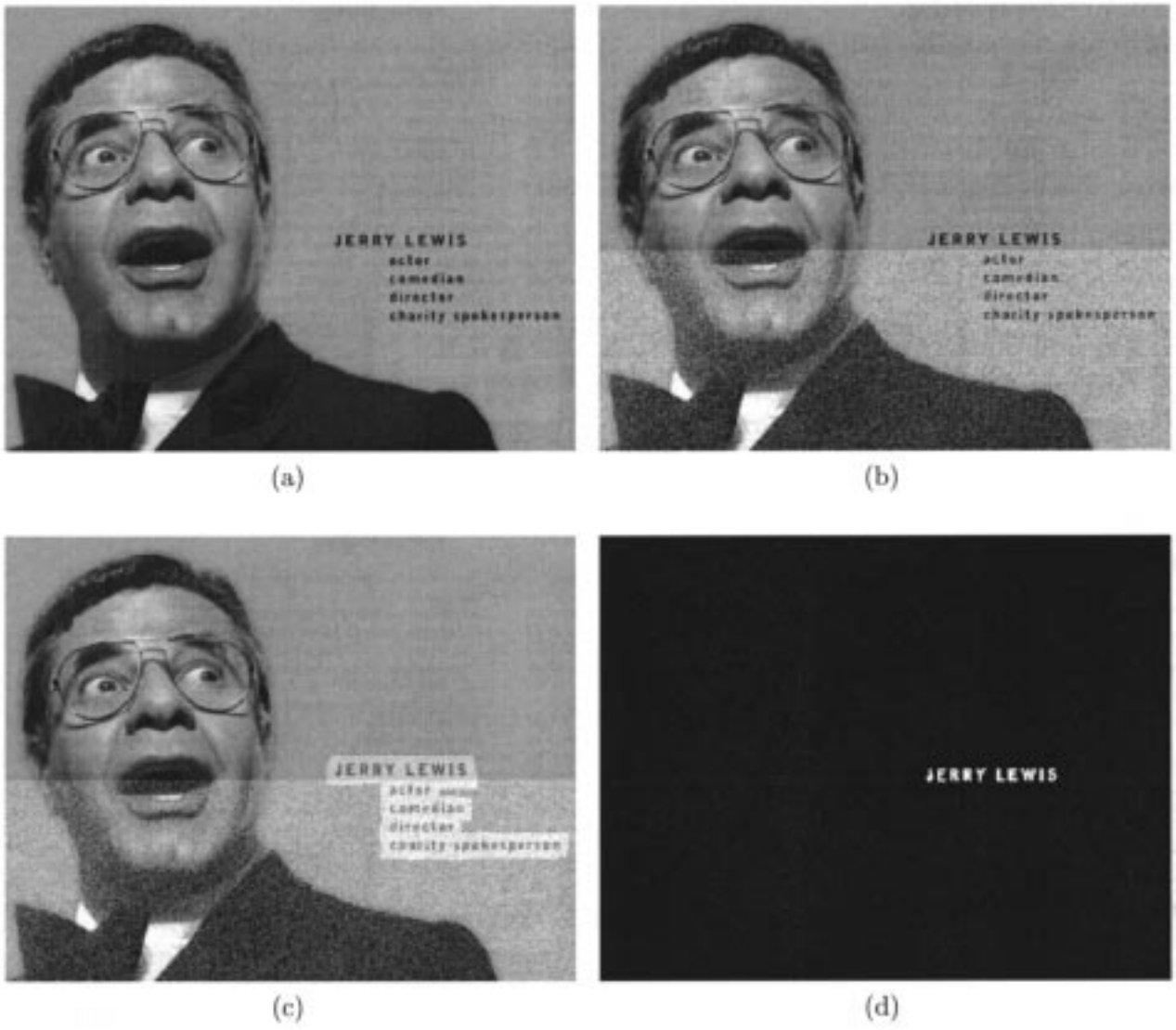

Fig. 6. Text extraction from the Jerry image with noise. (a) The intensity image Y of the original Jerry image with text. (b) Noise added to bottom part of the image in (a) with $\mathrm{SNR} = 10$ dB. (c) Results of the proposed algorithm; the extracted text block marked in the Jerry image. (d) Results of the algorithm of [9].

the text regions that satisfy (1), have been successfully extracted irrespective of the text orientation and without the need for a skew-correction step. Too small and too large text components are considered as noise and graphics, respectively, based on the employed size constraint (1). The existing algorithms of [7] and [9] fail to extract a text string if it is not horizontally aligned as shown in Fig. 5(c) and (d). Other existing techniques [2], [5], [20] for text extraction make use of skew-correction schemes to detect rotated text strings.

In addition, since existing text extraction techniques [9] are sensitive to noise, results are presented in Fig. 6 to illustrate the performance of the proposed algorithm in the presence of noise. Fig. 6(b) shows a noisy version of the Jerry image of Fig. 6(a) with noise added to the bottom part of the image with $\mathrm{SNR} = 10$ dB. The proposed algorithm successfully extracted the text region as shown in Fig. 6(c). Fig. 6(d) shows that the connected component-based algorithm [9] failed to extract the text in the noisy part of the image. For the proposed algorithm, it has been observed that text regions are successfully detected and no nontext regions are misclassified as long as the $\mathrm{SNR} \geq 10$ dB, which illustrates the robustness of the algorithm to noise since it is able to successfully execute even at low SNRs. If the SNR is less than 10 dB, text regions are still successfully detected but some nontext regions can be misclassified as text.

IV. Conclusion

In this paper, a novel technique for morphological text extraction from images is presented. The proposed method is robust to noise, insensitive to skew and text orientation, free from artifacts that are usually introduced by both fixed/optimal global thresholding and fixed-size block-based local thresholding. Generally, the output of the text extraction algorithm is fed to an OCR system to recognize the contained information. So, the main objective of the text extraction algorithm is to reduce the number of false text-candidate regions that may be fed to the OCR. Incorporating an OCR algorithm with the proposed morphological text extraction method yields a useful system for text analysis in images.

References

[1] K. C. Fan, L. S. Wang, and Y. K. Wang, "Page segmentation and identification for intelligent signal processing," *Signal Process.*, vol. 45, pp. 329–346, 1995.

[2] Z. Lu, "Detection of text region from digital engineering drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 431–439, Apr. 1998.

[3] Y. K. Ham, M. S. Kang, H. K. Chung, and R. H. Park, "Recognition of raised characters for automatic classification of rubber tires," *Opt. Eng.*, vol. 34, pp. 102–108, Jan. 1995.

[4] R. Lienhart, "Indexing and retrieval of digital video sequences based on automatic text recognition," in *Proc. ACM Int. Conf.*, Boston, MA, Nov. 1996.
[5] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 294–308, Mar. 1998.
[6] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 215–220, Feb. 1994.
[7] H. M. Suen and J. F. Wang, "Text string extraction from images of colorpriented documents," *Proc. Inst. Elect. Eng. Vis., Image, Signal Process.*, vol. 143, no. 4, pp. 210–216, 1996.
[8] L. Wang and T. Pavlidis, "Direct gray-scale extraction of features for character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1053–1067, Oct. 1993.
[9] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," *Pattern Recognit.*, vol. 28, no. 10, pp. 1523–1535, 1995.
[10] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Comput. Graph. Image Process.*, vol. 20, pp. 375–390, 1982.
[11] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.
[12] D. S. Bloomberg, "Multiresolution morphological analysis of document images," *Proc. SPIE Visual Communication Image Processing*, vol. 1818, pp. 648–662, 1992.
[13] R. Kasturi and M. M. Trivedi, *Images Analysis Applications*. New York: Marcel Dekker, 1990.
[14] K. K. Chin and J. Saniie, "Morphological processing for feature extraction," *Proc. SPIE*, vol. 2030, pp. 288–302, 1993.
[15] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Comput. Vis. Graph. Image Processing*, vol. 52, pp. 171–190, 1990.
[16] Y. M. Y. Hasan and L. J. Karam, "Morphological reversible contour representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 227–240, Mar. 2000.
[17] S. E. Umbaugh, Y. Wei, and M. Zuke, "Feature extraction in image analysis," *IEEE Eng. Med. Biol. Mag.*, vol. 16, no. 4, pp. 62–73, 1997.
[18] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[19] P. Yang and P. Maragos, "Morphological systems for character recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1993, pp. 97–100.
[20] B. Yu, A. Jain, and M. Mohiuddin, "Address block location on complex mail pieces," in *Proc. 4th Int. Conf. Document Analysis Recognition*, Ulm, Germany, 1997.

A Wavelet-Frame Based Image Force Model for Active Contouring Algorithms

Hsien-Hsun Wu, Jyh-Charn Liu, and Charles Chui

***Abstract*—This paper proposes a directional image force (DIF) for active contouring. DIF is the inner product of the zero crossing strength (ZCS) of wavelet frame coefficients, and the normal of a snake. By representing strength and orientation of edges at multiple resolution levels, DIF markedly improves the immunity of snakes to noise and convexity.**

I. Active Contouring Algorithms

The objective of active contouring, i.e., *snake* algorithms, is to find the closest contour around the natural boundary of an object. Terzopoulos and Kass [1] proposed the first snake algorithm, with several algorithms proposed later trying to solve various problems such as noise trap [5], snake initialization [2], energy function optimization [4], [5], image object extraction [6], concavity issues [3], [7], etc.

Manuscript received January 25, 1999; revised June 5, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kannan Ramchandran.

H.-H. Wu is with the Cytokinetics Inc., San Francisco, CA 94080 USA.
J.-C. Liu is with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: liu@cs.tamu.edu).
C. Chui is with the TeraLogic, Inc., San Jose, CA 95101 USA.

Publisher Item Identifier S 1057-7149(00)09395-7.

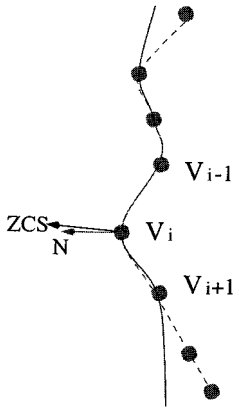

Fig. 1. Relationship between ZCS and the normal of the snake.

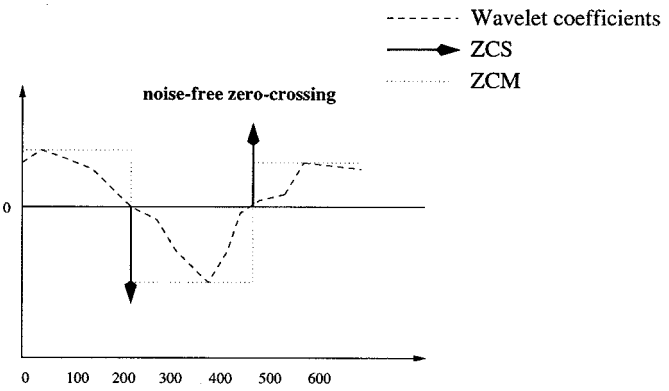

Fig. 2. Zero crossing representation.

Let the line segments that connect adjacent pairs of the n snaxels on a snake be denoted as $v(0),\ v(1),\ v(2),\ \cdots,\ v(n)$, where $|v(s)|$ is the arc length. Snake shape deformation is controlled by an *energy function* that consists of the *internal* and *external energy*,

$$E_{snake}(v(s)) = \int_0^1 [E_{int}(v(s)) + E_{ext}(v(s))]\, ds. \qquad (1)$$

The internal energy is commonly defined as $E_{int}(v(s)) = 1/2(w_1(s)|v'(s)|^2 + w_2(s)|v''(s)|^2)$, where $v'(s) \equiv (\partial v(s)/\partial s)$, and $v''(s) \equiv (\partial^2 v(s)/\partial s^2)$. $E_{ext}(v(s))$ is usually derived from lines or corners surrounding the snake. Ideally, a snake should stop the deformation process when its energy function is minimized, or $(\partial E_{snake}/\partial v) = 0$, where v is the discretized snake, provided that the spot noise does not trap a snaxel.

In this study, we adopt the expansion (ballooning) approach, where the external energy is the weighted sum of the image force and the balloon force [5]: $\int_0^1 E_{ext}(v(s)) = \int_0^1 E_{image}(v(s)) + \int_0^1 E_{balloon}(v(s))$. By using Green's theory, we have $\int_0^1 E_{balloon}(v(s)) = w_4(s) \iint_R dA(s)$, where R is the snake region and $w_4(s)$ is a weight factor. This newly proposed image force, called the *directional image force* (DIF), is the inner product of ZCS and the contour normal (see Fig. 1), $E_{image\_dir}(v(s)) = w_3(s)\overrightarrow{N(v(s))} \bullet \overrightarrow{ZCS(v(s))}$, where $w_3(s)$ is a weight constant, and ZCS is the *polarized strength* of a *zero*