Pergamon

Pattern Recognition, Vol. 28, No. 10, pp. 1523–1535, 1995
Elsevier Science Ltd
Pattern Recognition Society
Printed in Great Britain.
0031-3203/95 $9.50 + .00

Research Papers

0031-3203(95)00030-5

LOCATING TEXT IN COMPLEX COLOR IMAGES

YU ZHONG, KALLE KARU and ANIL K. JAIN*
Department of Computer Science, Michigan State University, East Lansing, MI 48824, U.S.A.

(*Received* 23 *November* 1994; *in revised form* 3 *February* 1995; *received for publication* 9 *March* 1995)

Abstract—There is a substantial interest in retrieving images from a large database using the textual information contained in the images. An algorithm which will automatically locate the textual regions in the input image will facilitate this task; the optical character recognizer can then be applied to only those regions of the image which contain text. We present two methods for automatically locating text in complex color images. The first method segments the image into connected components with uniform color, and uses several heuristics (size, alignment, proximity) to select the components which are likely to contain character(s) belonging to the text. The second method computes the local spatial variation in the gray-scale image, and locates text in regions with high variance. A combination of the two approaches is shown to be more effective than the individual methods. The proposed methods have been used to locate text in compact disc (CD) and book cover images, as well as in the images of traffic scenes captured by a video camera. Initial results are encouraging and suggest that these algorithms can be used in image retrieval applications.

Locating text Optical character recognition Color segmentation Spatial variance

1. INTRODUCTION

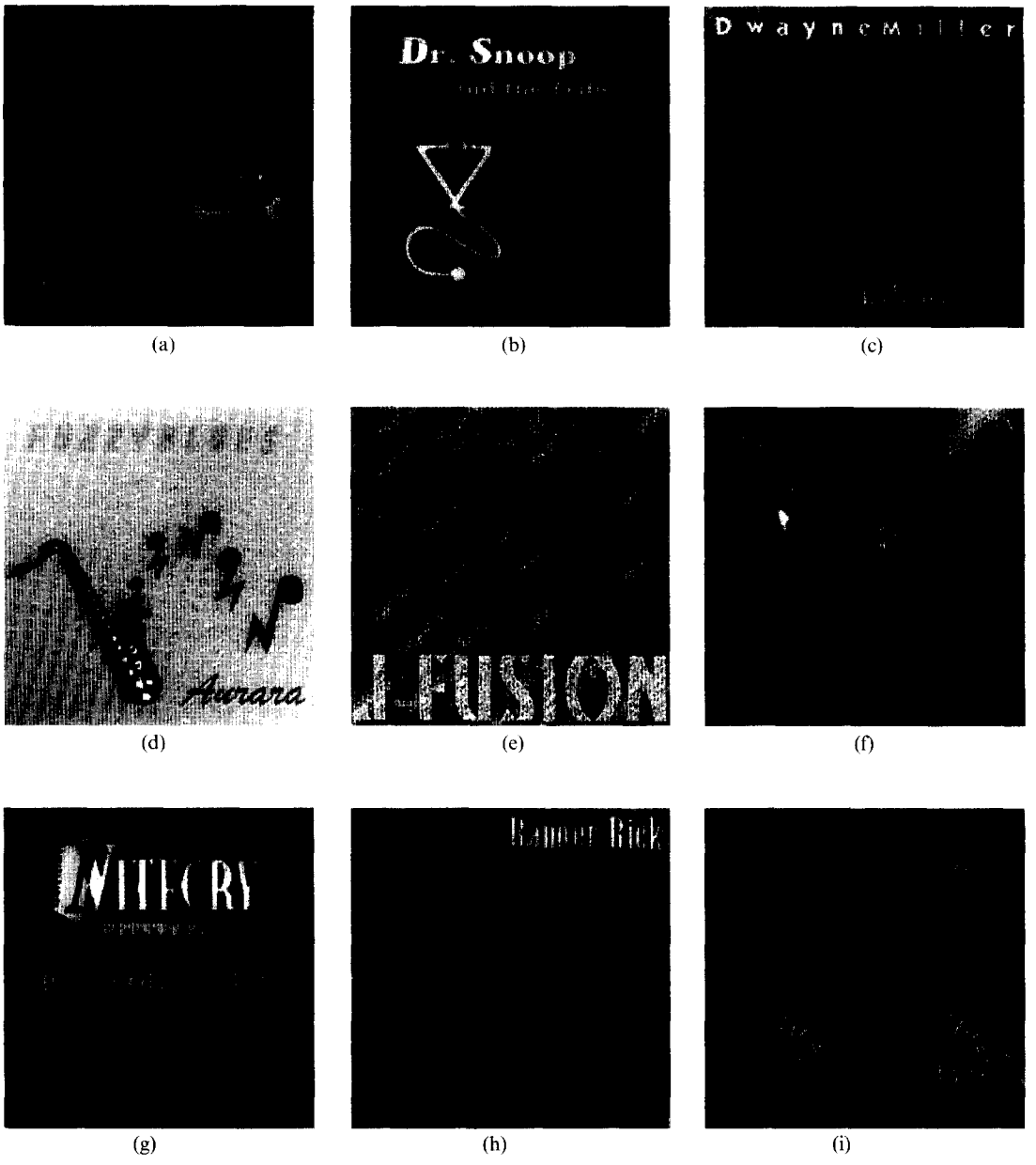
The purpose of a general image understanding system is to recognize objects in a complex scene, and to deduce relationships between the detected objects. While the universal image understanding system has not yet been constructed, many researchers have constrained the domain of interesting objects to be recognized, and devised methods for coping with these limited tasks. In this paper, we address the problem of automatically finding text in a complex color image. By a complex image we mean that the characters cannot be segmented by simple thresholding, and the color, size, font and orientation of the text is unknown.

Image segmentation into coherent regions is the first step before applying an object recognition method. For example, prior to using an optical character recognizer (OCR), the characters must be extracted from the image. Applications of locating and recognizing text include address localization on envelopes, scanning and understanding a printed document, identifying products (such as books, CDs, canned food, etc.) by reading the text on their covers or labels. Another application area of a text understanding system is in image database retrieval, where the capabilities of usual textual database systems are extended to image databases. Given a user's request, the system must retrieve all images in the database which match a text pattern (or generally, any kind of image pattern). A Query By Image Content (QBIC) database system has been reported,[1] and used for a practical application in art and history.[2] Considerable work has also been done with image databases of trademarks,[3,4] where it is important to check if a new trademark is very similar to any of the thousands of existing trademarks. Using text contained in an image to perform database retrieval is a special type of QBIC. For faster retrieval, the images in the database can be pre-processed by automatically extracting the text from them. Matching and retrieval are easier with text queries than with shape, texture or color attributes, but the complexity of extracting and recognizing the text balances this matching simplicity.

The methods presented in this paper have been primarily designed to extract text from CD cover images, some of which are shown in Fig. 1. The text on the covers may be printed in any font, even in cursive; the size of the text may vary from image to image, and what makes the task especially complicated, is the intermingling of differently colored objects in the background with the text whose color is not known *a priori*.

Previous research in recognizing text has focused mainly on character recognition. Text is often assumed to be printed in black on a white background, so that it can be extracted by thresholding the gray-scale image. Nevertheless, considerable research has been done in the area of page layout segmentation, where the main problem is to discriminate text areas from figures (halftones), both of them being relatively easy to extract from a white background. The difference between text and figure is obvious to a human reader. It is, however, difficult to formalize for an automatic system, and therefore various heuristics have been used in different page segmentation methods. We review here some of the heuristics because many of these ideas can be extended to the domain of complex color images.

* Author to whom correspondence should be addressed.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 1. CD cover images of size 356 × 378 pixels.

The most intuitive characteristics of text is its regularity. Printed text consists of characters with approximately the same size and the same line thickness. Characters are located about the same distance from each other, they are organized into lines of the same height, and the lines usually form columns that are justified from the two sides. Such regularities have been used implicitly[5,6] by considering text regions as having a certain texture with frequency components distinct form those of a gray-scale figure. The regular alignment of characters into lines and columns has been used more explicitly,[7] where text lines are assumed to form rectangles with distinct shape, and different lines are to be separated with white regions with no black pixels. These methods were successfully applied to obtain page layout of technical journals.

To extract characters from a more complex scene, the connected component approach has been used in reference (8). The image is first segmented into candidate character regions using adaptive thresholding and connected component analysis. A component can then be accepted as a character (or a part of a character) by applying additional heuristics, or simply classifying it with an OCR system which has a class for "non-character" reject option.

We note that in a general setting, it is impossible to extract text or characters from a complex scene without first recognizing it. However, feeding an OCR system

with all possible components in an input image is not reasonable because a typical image in Fig. 1, for example, results in thousands of components. Our approach is to segment the image into text/non-text regions as best as possible, and then let the OCR system do the detailed refinement. In other words, we would like to find all text areas and as few spurious non-text areas as possible, without actually classifying the characters.

In the next section, we describe two approaches for extracting text. The first approach is based on segmenting the color image into connected components of uniform color, and using various heuristics to reduce the number of candidate characters. The second method locates the text lines rather than individual characters by using the difference of spatial gray-value variance inside the text regions and in the background. A hybrid method which combines the two approaches is discussed at the end of the section. Section 3 describes experiments done with the CD and book cover images. We show that the variance-based approach generally works better than the connected components method, although the second method can be used to refine the result of the first method. In the final section, we summarize the paper and list the limitations of the proposed method.

2. LOCATING TEXT IN AN IMAGE

In this section we describe two different methods for extracting text from an input image. High-level algorithms of the two approaches are drawn in Figs 2 and 3. The first method divides the input color image into connected components, and examines these components to select the ones corresponding to text. The second method extracts text from a gray-scale image using regions with high spatial variance. In the following paragraphs, the two approaches are described for finding horizontally aligned text only. In principle, these methods can be easily modified to detect text in any orientation.

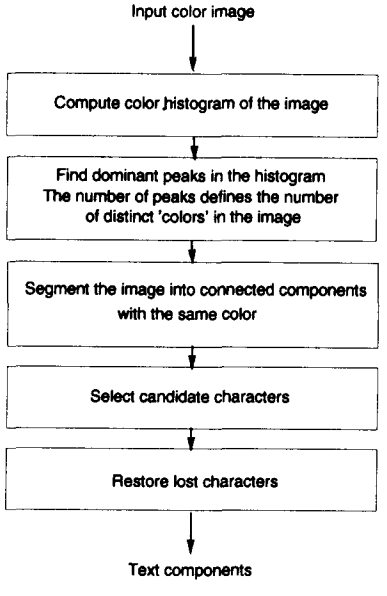


Fig. 2. Block diagram of the connected component method.

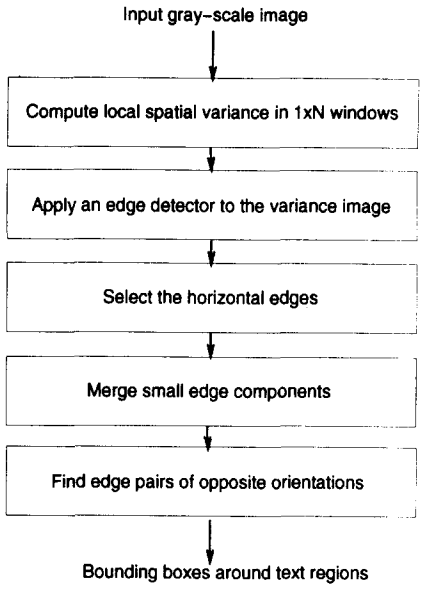


Fig. 3. Block diagram of the spatial variance based method.

2.1. *Connected component method*

The first step in segmenting an image is to define components with homogeneous color or gray-value. If we require that the color of an individual component be slowly varying, with sharp edges on its boundaries, then the basic algorithm for finding connected components can be applied. In that case, two neighboring pixels are in the same component if the difference between their color is less than a threshold. (To compute the distance between two colors we have experimented with both Intensity–Hue–Saturation (I–H–S) and Red–Green–Blue (R–G–B) color spaces. Although the I–H–S encoding can be directly related to the human color perception,[9] we have obtained better results with the Euclidean distance in the three-dimensional (3D) R–G–B space. This simple segmentation algorithm, however, does not work with most natural images where the characters have blurred boundaries, and are, therefore, connected with the background. A more discriminative method is to divide the image into components of constant color. The image segmentation can now be done by taking one color (a triple of the R–G–B values), and finding all connected components with this color. Due to the large number of possible colors, we first quantize the color space into a few prototype colors. The prototypes are found as local maxima in a smoothed color histogram of the input image, all other colors are then assigned to the nearest prototype. In most of the images used in our experiments, this results in about 5–500 peaks in the histogram, and the components that seem different to the human eye are quantized into different classes. Image segmen-

tation can then be done with the basic connected component algorithm, by connecting two neighboring pixels if they are assigned to the same color prototype.

The color segmentation phase results in approximately 1000–25,000 components, and the most crucial step is to select the correct components (those corresponding to the text). We have used the following three heuristics to remove the non-character components:

(1) The area of the component must be between two threshold values—*Minarea*, *Maxarea*.

(2) The diameter of the component must be less than *Maxdiameter*.

(3) A text line consists of at least three characters with the same color that are aligned along the top or bottom line (minimum or maximum row coordinate).

The actual parameter values that we have used in the experiments were *Minarea* = 100 pixels, *Maxarea* = 500 pixels, *Maxdiameter* = 100 pixels. The first two conditions can be checked by computing the area and the bounding box coordinates for each component. To satisfy the third requirement, we check for each component if its nearby components of the same color have the same minimum or maximum row coordinate value.

The thresholds in the heuristics can be selected so that all the retained components correspond to some valid character. In order to restore some of the characters (components) which were removed by mistake, we can use the information from the retained "good" (character) components. For each "bad" component which satisfies the first two conditions listed above (with slightly relaxed threshold values), we check if there is another "good" component with the same color, such that the row-centroid of the "bad" component is between the minimum and the maximum row coordinates of the "good" one. This last postprocessing step restores almost all the characters in a text line, given that at least one character in the line was found before.

It is clear that the above algorithm works only with horizontally aligned printed characters that have a distinct color from the background. Its main shortcoming is the inability to find touching characters, especially in cursive printed text. The requirement that the characters be separated from each other seems too restrictive. When we read some text, we do not segment it into connected components at the first glance, although horizontal alignment of the text seems to be important for fast recognition.

2.2. *Spatial variance method*

The second method was designed to detect handwritten text, provided that it is aligned horizontally. The main idea of this algorithm can be explained by considering this page of the manuscript. If we compute the spatial variance along each horizontal line over the whole image, we see that regions with high variance correspond to text lines, and regions with low variance correspond to white regions between the text lines. Moreover, there are steep edges corresponding to the minimal and maximal row coordinates of small letters, such as "a", "o", and "m". Text lines can then be found by extracting the rows between two sharp edges of the spatial variance—one edge rising and the other falling.

The same basic method can be applied to a more complex image, assuming that the spatial variance in the background is lower than in the text. Since we need to locate both the row coordinates of a text line and the column coordinates of its beginning and end, the spatial variance must be computed for each pixel over a local neighborhood in the horizontal direction (in a window of size 1×21 pixels). This results in an image of horizontal spatial variances with the same size as the input image. From this image we need to find significant horizontal edges and then pair the edges with opposite directions into lower and upper boundaries of a text line. In principle, any edge detector can be used to locate the edges. In our experiments, we have used Canny edge detector, which locates edges more accurately than other, simpler methods. Since we are currently interested only in horizontal text lines, we select those edges which have horizontal direction ($\pm 15°$).

Before finding pairs of edges with opposite directions, small horizontal edge components need to be merged into longer lines. The edge merging is performed by first finding the connected components of the edge pixels. The components which have a large vertical spread (more than 10 pixels) are not good candidates for upper and lower boundaries of a text line, and can be removed. Among the remaining lines, we find the ones which have almost the same row coordinates (± 3 pixels), and merge them into a single line. The merged edges, of course, must have the same orientations.

The final step in the algorithm is to group together pairs of lines with opposite orientations. The huge number of all possible groupings is reduced by using the following heuristics:

(1) A significant portion of the two lines must overlap when they are projected vertically (in the experiments, we required the overlap to be at least 1/2 of each line).

(2) A rising edge must be above the falling edge.

(3) There must be no other line between the two edges in a pair.

(4) The distance between the upper and the lower edge should not be very small (less than 10 pixels) or very large (more than 40 pixels). This heuristic constrains the character size.

From a pair of lines, it is a simple matter to construct the smallest rectangle which contains the two "paired" lines in its upper and lower edges. All such rectangles form the output of the algorithm—the bounding boxes around the candidate text regions.

2.3. *Hybrid algorithm*

The second method generally gives better results than the connected component analysis when the

characters are connected or not well-separated from the background. The output of the algorithm—bounding boxes—needs some preprocessing by an OCR system because the latter usually expects as input segmented characters. Another drawback of the spatial variance method is that the characters that extend below the baseline or above the other characters ("g", "p", "t", for example) are sometimes cut into halves. These two disadvantages of the algorithm can be eliminated by using the connected component approach after locating the bounding boxes, to fill in the character regions extending beyond the boxes, and to segment the text inside the boxes into separate characters, if possible.

The following steps merge the two methods into one algorithm (see also Fig. 4 for a block diagram):

(1) Find the bounding boxes around the text components in the input image using the spatial variance method.

(2) Use the first method to find connected components both in the whole image and in each box separately.

(3) For each box, determine the color of the text inside it as the color whose proportion drops most sharply when the box size is increased two-fold.

(4) Extract text inside each box using the text color determined in step 3.

(5) Fill in the text components extending beyond the bounding boxes. For each text component inside a box, extend it to the component in the entire image. If the new component does not extend far from the box, call it a text component.

(6) Include the beginning and ending characters that are outside the boxes. For each component in the entire image that has a proper size, check if its row-centroid is between the upper and the lower boundaries of some bounding box in which the text color is the same as the color of the component.

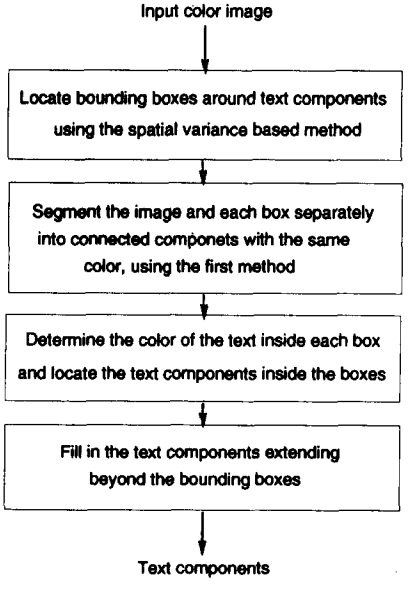


Fig. 4. Block diagram of the hybrid method.

Before presenting the experimental results, we mention that although the two methods were presented to detect horizontally aligned text only, both of the methods can be extended to find text in any orientation, provided that it is aligned between two parallel lines. Such a generalized algorithm, however, is expected to given more spurious non-text regions.

3. EXPERIMENTAL RESULTS

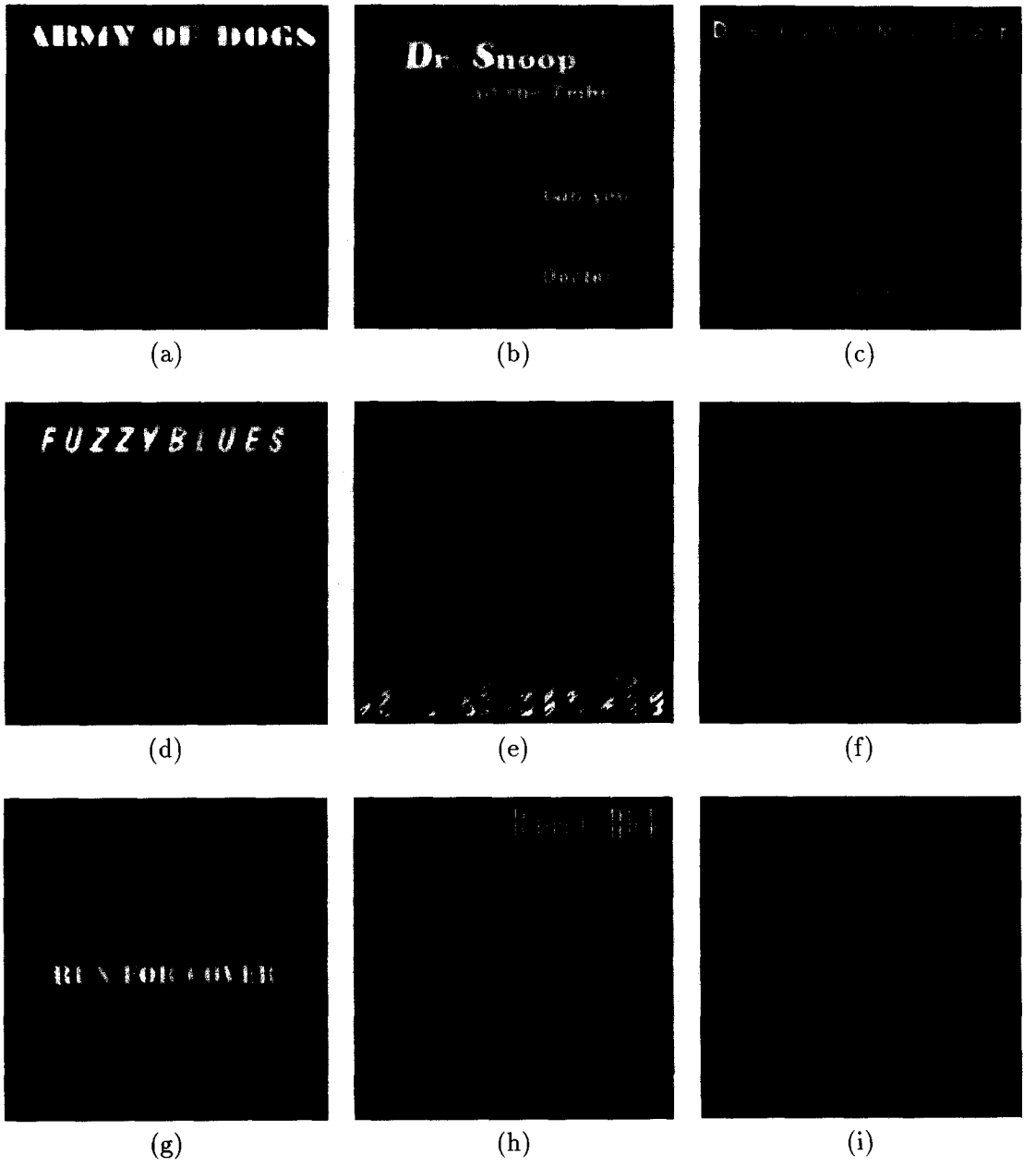
As mentioned earlier, the two methods were designed primarily to locate text in color CD cover images. In Fig. 1 we show a sample of these images. The images in this figure are not representative, because a majority of CDs contain horizontally aligned printed text that is relatively easy to detect. The samples in Fig. 1 have been selected to show a variety of possible texts, including handwritten text, non-horizontally aligned characters, and text with varying color. We will show, using these sample images, the situations where the proposed methods work well and where they fall short.

The two algorithms require several parameters that must be set empirically, and which can be tuned for a class of image. In the experiments reported here, we have tried to select one set of parameters which gives acceptable results on a variety of images. Figure 5 shows the result of applying the first method to all the test images. As can be seen, if the text is printed with a distinct color and is horizontally aligned, then the algorithm finds the characters well. It misses text which does not have a dominant color (a), handwritten text (d, f, i), text that is not horizontally aligned (e, f), and text which is connected to background (g). Also, in image (b) the short words "DOG" and "the" are not detected.

Figure 6 contains the results of applying the second method to the input CD cover images. The bounding boxes are shown as white rectangles on the input images. Almost all the text regions are bounded by the boxes except the vertical text in (f) and the scattered characters in (e). In addition, there are a few boxes that do not contain any text in (b, f, g, h).

In Fig. 7 the combination of the two methods is applied to the CD cover images. In addition to displaying the text components in each bounding box, we have also shown enlarged boxes around the detected text. It can be seen that the text which has a uniform color and is bounded by some box can be located accurately. A few spurious boxes produce some non-text components, which can be easily removed by an OCR system. In Fig. 8 we have overlaid the boxes located using the combined method on the original images.

Figure 9 shows color images of three book covers. The results of applying the spatial variance based method and the combination of the two methods are shown in Figs 10, 11 and 12. Generally the text components have been located correctly, except some text



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 5. Results of applying the connected component method to the CD cover images.

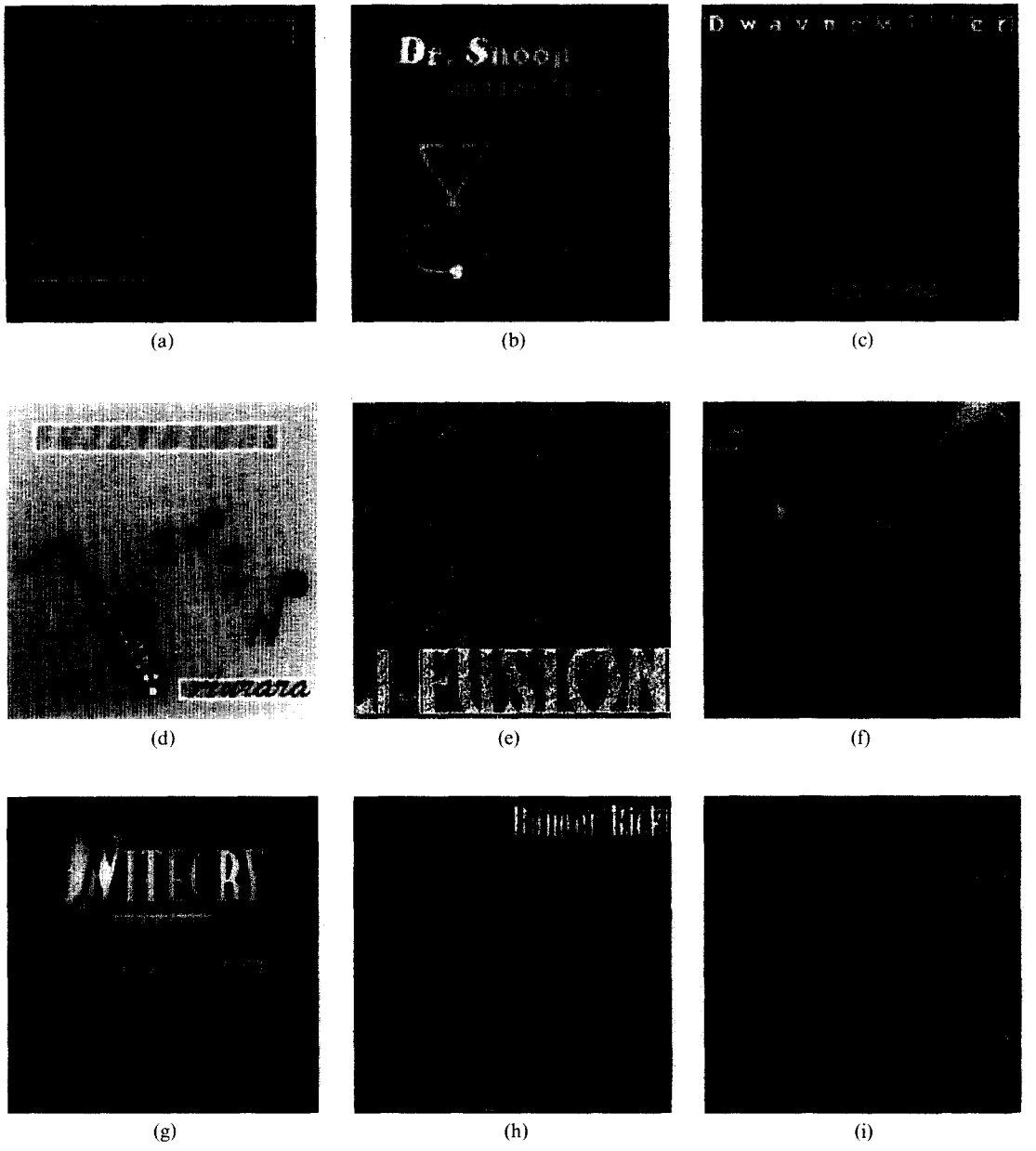


Fig. 6. Results of applying the spatial variance based method to the CD cover images.

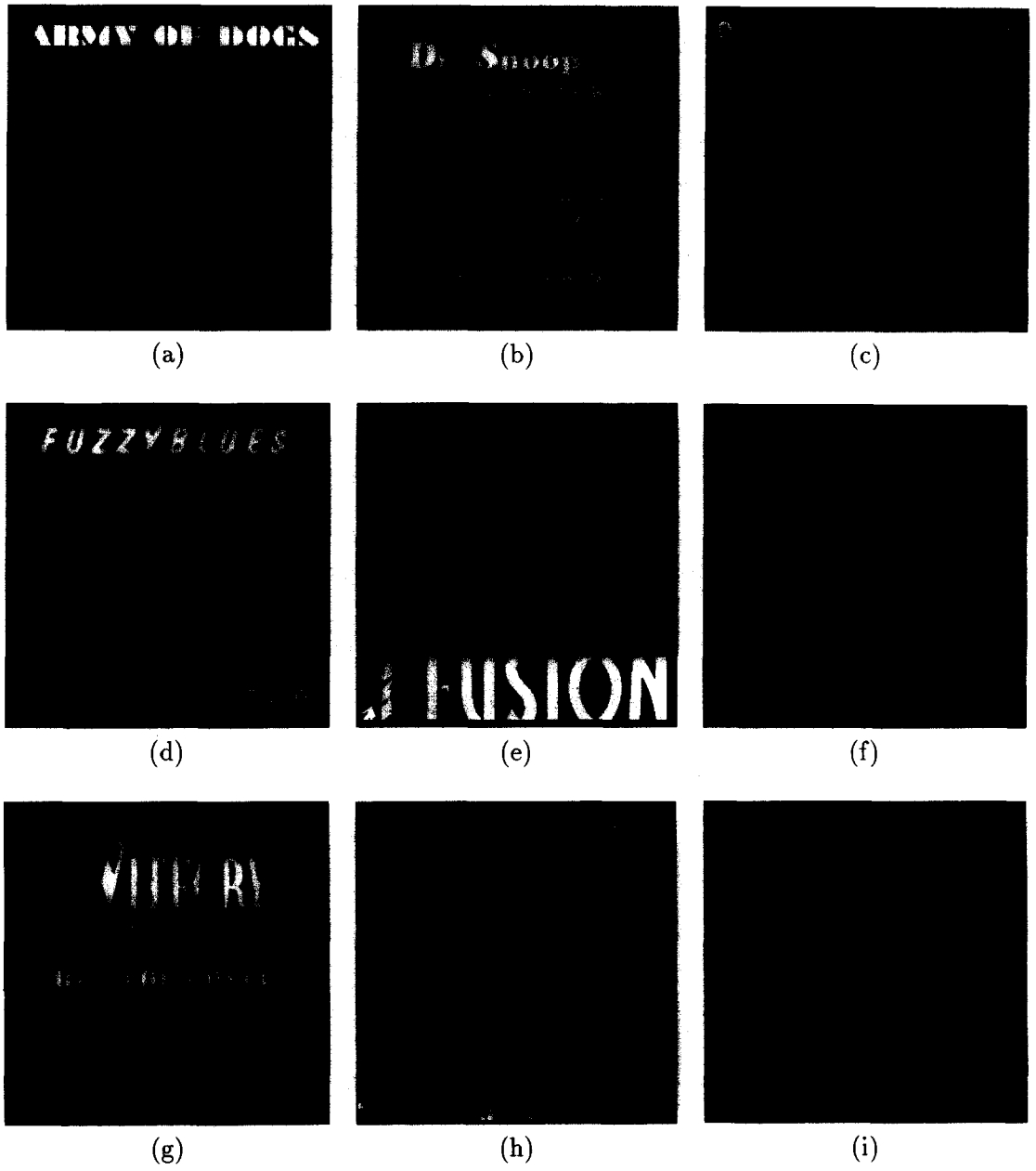
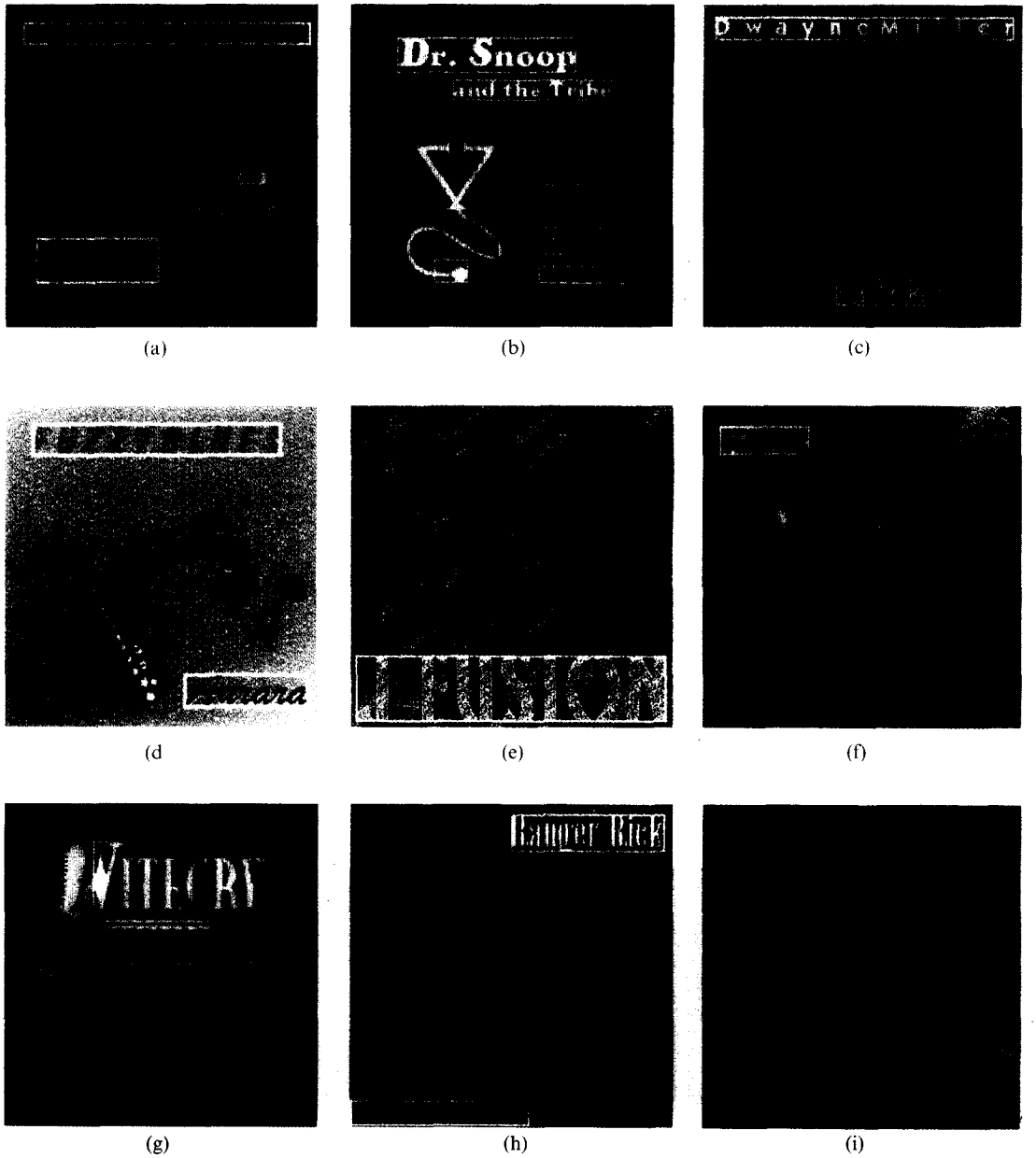


Fig. 7. Results of applying the hybrid method to the CD cover images.

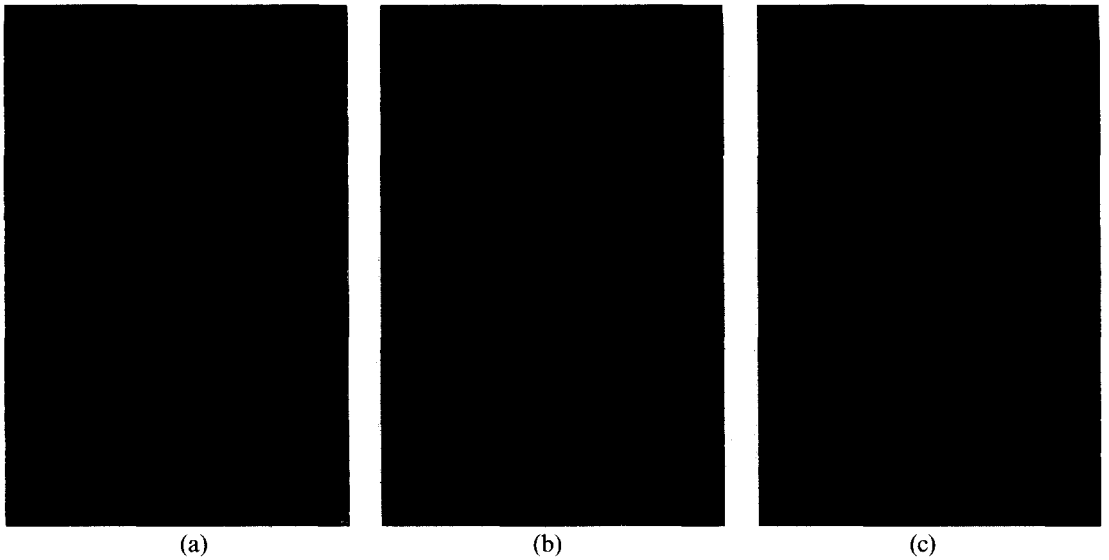


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 8. Boxes found using the hybrid method, shown on the original images.



(a) (b) (c)

Fig. 9. Examples of book covers with size 468 × 376 pixels.

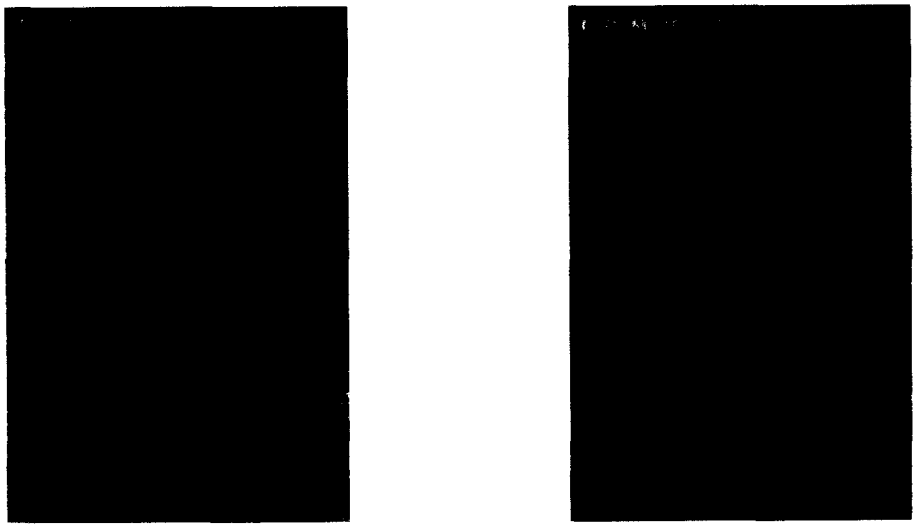


Fig. 10. Results of applying the spatial variance based method and the hybrid method to the first book cover image (Fig. 9).

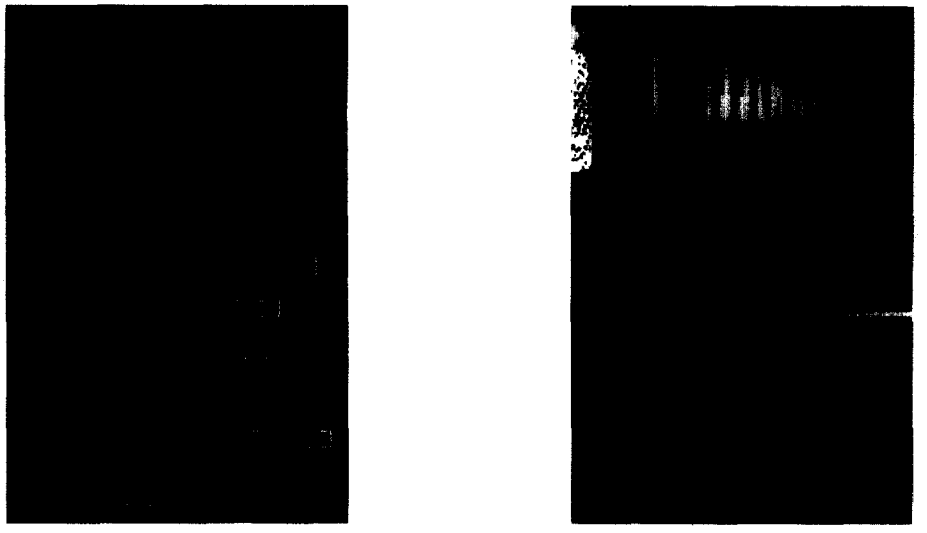


Fig. 11. Results of applying the spatial variance based method and the hybrid method to the second book cover image (Fig. 9).

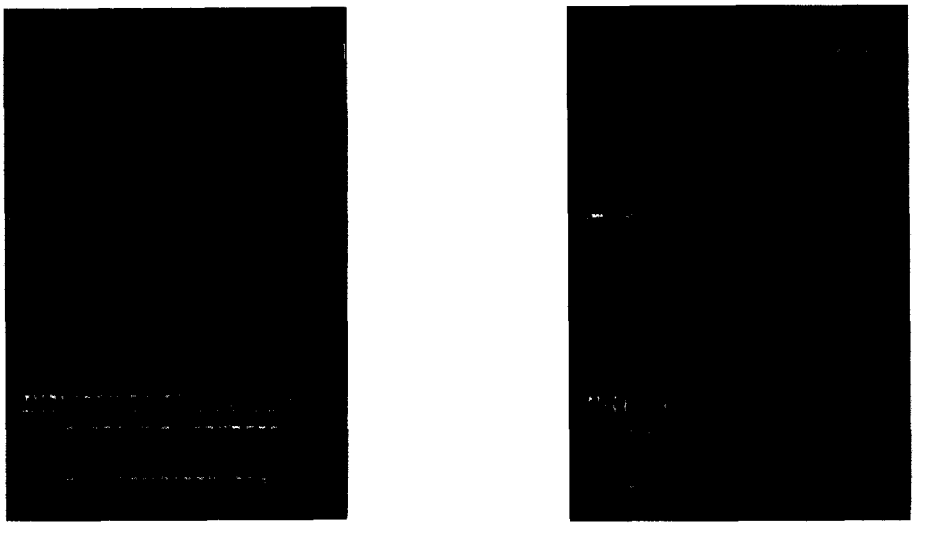


Fig. 12. Results of applying the spatial variance based method and the hybrid method to the third book cover image (Fig. 9).

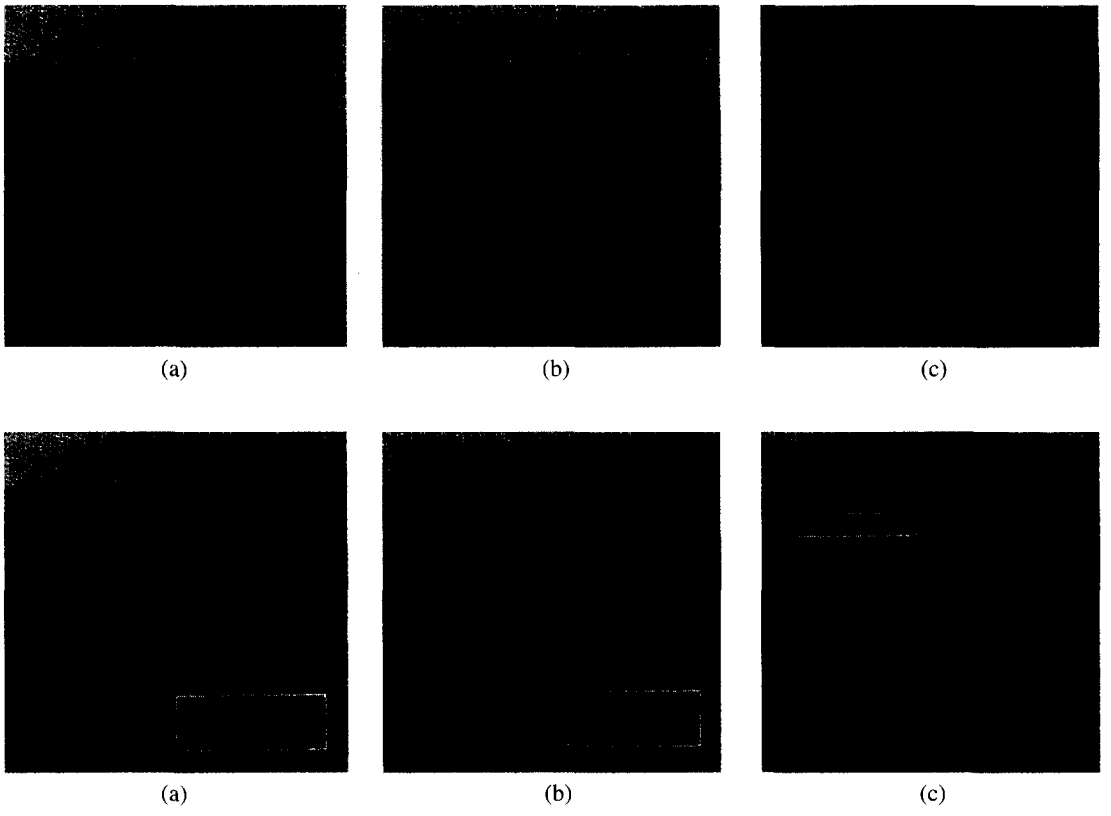


Fig. 13. Car images with size 256 × 256 pixels and the results of applying the spatial variance method.

with very low contrast. In Fig. 11, the combined method failed to recognize the correct text color in two boxes.

Figure 13 contains three car images (three frames from a video sequence of a traffic scene) and the results of applying the spatial variance based method to these images. We did not use the connected component or the hybrid method because the color of the characters and the background is almost the same in these images.

All the three algorithms for locating text can be implemented to execute fast. The first method requires one scan over the whole image to construct the color histogram, and two other scans to find connected components. Additional computations for selecting candidate components depend on how effective data structures can be designed, so that each component is compared with only its nearby neighbors. The second method, in addition to performing edge detection, requires no more than two scans over the image to compute spatial variance, and the same number of scans to perform the connected component analysis of the edge pixels. The processing speeds of the current implementations of these algorithms on a typical 256 × 256 image are given in Tables 1 and 2. Although the second method was designed for gray-scale images, it can be extended (and possibly improved) to color images by finding a suitable method for color edge detection, and by defining the spatial variance in a color image.

Table 1. Processing time of the connected component method on a 256 × 256 color image on SPARCstation 20

Operation	Time (s)
Compute color histogram	1.8
Find connected components	2.4
Select candidate components	1.2
Restore lost characters	0.7

Table 2. Processing time of the spatial variance method on a 256 × 256 color image on SPARCstation 20

Operation	Time (s)
Compute spatial variance	1.8
Apply Canny edge detector	4.0
Merge small edge components	0.5
Find matching edge pairs	0.3

4. CONCLUSION

Two methods for extracting text from a complex image are described in this paper. The approaches are based on certain heuristics, and the algorithms can, therefore, not be expected to find text in all possible situations. The most serious shortcomings are in not

locating the text which is not well separated from the background (this means that the color of the text and background is similar, or that the contrast of the text is low), and characters which are not aligned, or which have differently colored components. Nevertheless, the methods give good results on a variety of test images. The algorithms are relatively robust to variations in font, color, or size of the text. While neither of the two methods works perfectly with all types of images, acceptable performance can be obtained when the domain of the input images is restricted to video images, CD or book covers, for example. A combination of the two methods was shown to locate text more accurately than either of the two methods separately.

REFERENCES

1. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, Efficient and effective querying by image content, *J. Intell. Inform. Systems* **3**, 231–262 (1994).
2. B. Holt and L. Hartwick, Visual image retrieval for applications in art and art history, *Proc. of SPIE* **2185**, 70–81 (1984).
3. G. Cortelazzo, G. A. Mian, G. Vezzi and P. Zamperoni, Trademark shapes description by string-matching techniques, *Pattern Recognition* **27**, 1005–1018 (1994).
4. B. M. Mehtre, M. S. Kankanhalli, A. D. Narasimhalu and G. C. Man, Color matching for image retrieval, Institute of Systems Science, National University of Singapore, Technical report TR94-138-0.
5. A. K. Jain and S. Bhattacharjee, Text segmentation using gabor filters for automatic document processing, *Mach. Vision Applic.* **5**, 169–184 (1992).
6. A. K. Jain and Y. Zhong, Page layout segmentation based on texture analysis, (submitted).
7. T. Pavlidis and J. Zhou, Page segmentation and classification. *Comput. Vision Graphics Image Process.* **54**, 484–496 (1992).
8. J. Ohya, A. Shio and S. Akamatsu, Recognizing characters in scene images, *IEEE Trans.* **PAMI-16**, 214–224 (1994).
9. Q.-T. Luong, Color in computer vision, *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau and P. S. P. Wang eds. 311–368. World Scientific Publishing Company (1993).

About the Author—ANIL JAIN received a B.Tech. degree in 1969 from the Indian Institute of Technology, Kanpur, and the M.S. and Ph.D. degrees in electrical engineering from the Ohio State University, in 1970 and 1973, respectively. He joined the faculty of Michigan State University in 1974, where he currently holds the position of University Distinguished Professor in the Department of Computer Science. Dr Jain served as Program Director of the Intelligent Systems Program at the National Science Foundation (1980–1981), and has held visiting appointments at Delft Technical University, Holland, Norwegian Computing Center, Oslo and Tata Research Development and Design Center, Pune, India. He has also been a consultant to several industrial, government and international organizations. His current research interests are computer vision, image processing, and pattern recognition.

Dr Jain has made significant contributions and published a large number of papers on the following topics: statistical pattern recognition, exploratory pattern analysis, Markov random fields, texture analysis, interpretation of range images, and 3D object recognition. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979 and 1992 from the Pattern Recognition Society. Dr Jain served as the Editor-in-Chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1991–1994), and currently serves on the editorial boards of *Pattern Recognition* journal, *Pattern Recognition Letters*, *Journal of Mathematical Imaging*, *Journal of Applied Intelligence*, and *IEEE Transactions on Neural Networks*. He is the co-author of *Algorithms for Clustering Data*, Prentice-Hall, 1988, has edited the book *Real-Time Object Measurement and Classification*, Springer-Verlag, 1988, and has co-edited the books, *Analysis and Interpretation of Range Images*, Springer-Verlag, 1989, *Neural Networks and Statistical Pattern Recognition*, North-Holland, 1991, *Markov Random Fields: Theory and Applications*, Academic Press, 1993, and *3D Object Recognition*, Elsevier, 1993.

Dr Jain is a Fellow of the IEEE. He was the Co-General Chairman of the 11th International Conference on Pattern Recognition, Hague (1992), General Chairman of the IEEE Workshop on Interpretation of 3D Scenes, Austin (1989), Director of the NATO Advanced Research Workshop on Real-time Object Measurement and Classification, Maratea (1987), and co-directed NSF supported Workshops on "Future Research Directions in Computer Vision", Maui (1991), "Theory and Applications of Markov Random Fields", San Diego (1989) and "Range Image Understanding", East Lansing (1988). Dr Jain was a member of the IEEE Publications Board (1988–1990) and served as the Distinguished Visitor of the IEEE Computer Society (1988–90).

About the Author—KALLE KARU received a Diploma from Tartu University, Estonia, in 1993. He is currently a graduate student at Michigan State University. His current research interests are computer vision and image processing.

About the Author—YU ZHONG received the B.S. and M.S. degrees in computer science and engineering from Zhejiang University, Hangzhou, China in 1988 and 1991, the M.S. degree in statistics from Simon Fraser University, Burnaby, Canada, in 1993. She is currently a Ph.D. student at Michigan State University. Her research interests include image processing and machine vision.