Springer Link

ICAPR: International Conference on Pattern Recognition and Image Analysis

Pattern Recognition and Data Mining

Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I

- Editors
- (view affiliations)

- Sameer Singh
- Maneesha Singh
- Chid Apte
- Petra Perner

Conference proceedings **ICAPR 2005**

- 62 Citations
- 84 Readers
- 53k Downloads

Part of the Lecture Notes in Computer Science book series (LNCS, volume 3686)

Table of contents

Previous page of 4

Previous page of 4

Other volumes

1. Pattern Recognition and Data Mining
   Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I
2. Pattern Recognition and Image Analysis
   Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part II

About these proceedings

Keywords

Augmented Reality biometrics data mining image analysis image processing learning pattern recognition

Editors and affiliations

- Sameer Singh (1)
- Maneesha Singh (2)
- Chid Apte (3)
- Petra Perner (4)

1. Research School of Infomatics, Loughborough, UK
2. ATR Lab, Research School of Informatics, University of Loughborough, Loughborough, UK
3. IBM Corporation, New York, United States
4. Institute of Computer Vision and applied Computer Sciences, IBaI, Germany

Bibliographic information

- DOI (Digital Object Identifier) https://doi.org/10.1007/11551188
- Copyright Information Springer-Verlag Berlin Heidelberg 2005
- Publisher Name Springer, Berlin, Heidelberg
- eBook Packages Computer Science
- Print ISBN 978-3-540-28757-5
- Online ISBN 978-3-540-28758-2
- Series Print ISSN 0302-9743
- Series Online ISSN 1611-3349

A Novel Approach for Text Detection in Images Using Structural Features

H. Tran[1,2], A. Lux[1], H.L. Nguyen T[2], and A. Boucher[3]

[1] Institut National Polytechnique de Grenoble,
Laboratory GRAVIR, INRIA Rhone-Alpes, France
[2] Hanoi University of Technology,
International Research Center MICA, Hanoi VietNam
[3] Institut de la Francophonie pour l'Informatique
thi-thanh-hai.tran@inrialpes.fr

Abstract. We propose a novel approach for finding text in images by using ridges at several scales. A text string is modelled by a ridge at a coarse scale representing its center line and numerous short ridges at a smaller scale representing the skeletons of characters. Skeleton ridges have to satisfy geometrical and spatial constraints such as the perpendicularity or non-parallelism to the central ridge. In this way, we obtain a hierarchical description of text strings, which can provide direct input to an OCR or a text analysis system. The proposed method does not depend on a particular alphabet, it works with a wide variety in size of characters and does not depend on orientation of text string. The experimental results show a good detection.

1 Introduction

The rapid growth of video data creates a need for efficient content-based browsing and retrieving systems. Text in various forms is frequently embedded into images to provide important information about the scene like names of people, titles, locations or date of an event in news video sequences, etc. Therefore, text should be detected for semantic understanding and image indexation. In the literature, text detection, localisation, and extraction are often used interchangeably. This paper is about the problem of detection and localisation. Text detection refers to the determination of the presence of text in a given image and text localisation is the process of determining the location of text in the image and generating bounding boxes around the text.

For text detection we need to define what text is. A text is an "alignment of characters", characters being letters or symbols from a set of signs which we do not specify in advance. In images, text can be characterised by a region of elongated shape band containing a large number of small strokes. The style and the size of characters can vary greatly from one text to another. In images of written documents background as well as text color are nearly uniform, the detection of text can easily be performed by thresholding the grayscale image. However, the task of automatic text detection in natural images or video frames

S. Singh et al. (Eds.): ICAPR 2005, LNCS 3686, pp. 627–635, 2005.

is more difficult due to the variety in size, orientation, color, and background complexity. A generic system for text extraction has to cope with these problems.

2 Methods of Text Detection in Images

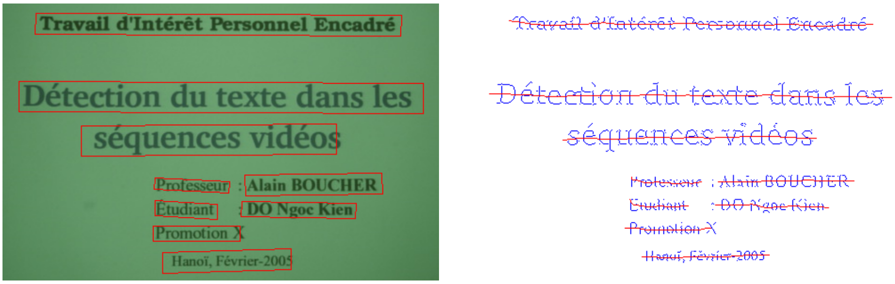
Approaches for detecting and localizing text in images in the literature can be classified into three categories: (1) bottom-up methods [6,8], (2) top-down methods [12,14] and (3) machine learning based top-down methods [5,7]. The first category extracts regions in image and then groups character regions into words by using geometrical constraints such as the size of the region, height and width ratio. These methods avoid explicit text detection but they are very sensitive to character size, noise and background complexity. In the second category, characters can be detected by exploiting the characteristics of vertical edge, texture, edge orientation and spatial properties. These methods are fast but give false alarms in case of complex background. The third category has been developed recently and receives much attention from researchers. The evaluation of machine learning based methods showed the best performance in comparison with other approaches [1]. The principle is to extract some characteristics like wavelets [5], statistical measures [2,3] or derivatives [7] from fixed-size blocks of pixels and classify the feature vectors into text or non-text using artificial neural networks. As usual with this kind of learning method, the quality of results depends on the quality of the training data and on the features which are fed into the learning machine.

3 Proposed Approach

The objective of our work presented in this paper is to construct an automatic text detector which is independent with respect to the size, the orientation and the color of characters and which is robust to noise and aliasing artifacts. We propose a new method of text detection in images that is based on a structural model of text and gives more reliable results than methods using purely local features like color and texture.

The structural features used here are ridges detected at several scales in the image. A ridge represents shape at a certain scale. Analyzing ridges in scale space permits to capture information about details as well as global shape. A line of text is considered as a structured object. At small scales we can clearly see the strokes. At lower resolution, the characters disappear and the text string forms an elongated cloud. This situation can be characterized by ridges at small scales representing skeletons of characters and at coarser scale representing the center line of the text string (figure 1). These properties are generic for many kinds of text (scene text, artificial text or targeted scene text), do not depend on the alphabet (e.g. latin characters, ideograms), and also apply for hand written text (figure 2).

The ridge detection operator is iso-symmetric so it can detect a text string as straight line or curve at any orientation (figure 2d). The multi-resolution computation detects a wide variety in text size. Unlike the multi-resolution approach proposed in [11,12], where candidate text is detected at each scale separately and requires an additional scale fusion stage, our work directly exploits the topological change of text over scale. In addition, analyzing the relation between scales and ridge lengths can predict the number of characters in a text line, and character dimensions.


Travail d'Intérêt Personnel Encadré
Détection du texte dans les
séquences vidéos
Professeur : Alain BOUCHER
Étudiant : DO Ngoc Kien
Promotion X
Hanoï, Février-2005

(a) (b)

Fig. 1. (a) Image of a slide; detected text regions are bounded by red rectangles. (b) Ridges detected at two levels $\sigma_1 = 2\sqrt{2}$ (blue) and $\sigma_2 = 16$ (red): red lines represent the center lines of text strings, blue lines represent skeletons of characters.

The rest of this paper is organized as follows: In section 4, we present briefly the definition of ridge and explain the representation of text line based on ridges. We then analyze in detail the constraints that a text region must satisfy to be discriminated from a non-text region. Some experimental results and conclusions will be shown in sections 5 and 6 respectively.

4 Text Detection Based on Ridges

This section explains the method for finding text regions in images based on ridges. It consists of 2 stages: (1) computing ridges in scale space and (2) classifying regions corresponding to ridges into 2 classes: text or non-text.

4.1 Computing Ridges at Multiple Scales

This section briefly explains ridge detection. For more technical details, see [9]. Given an image $\mathcal{I}(x, y)$ and its laplacian $\mathcal{L}(x, y)$ a point (x_r, y_r) is a *ridge point* if the value of its laplacian $\mathcal{L}(x_r, y_r)$ is a local maximum in the direction of the highest curvature; it is a *valley point* if the value of its laplacian $\mathcal{L}(x_r, y_r)$ is a local minimum. In the sequel, we use the term "ridge" to indicate these two types of points. Ridge points are invariant to image rotation and translation.

To detect ridge points, we compute the main curvatures and associated directions at each pixel using the eigenvalues and eigenvectors of the Hessian matrix [4]. We then link ridge points to form ridge lines by connected components analysis.

Scale space adds a third dimension σ to the image such that $I_\sigma(x,y)$ is the original image I smoothed by a Gaussian kernel with standard deviation σ. In our system, we use a discrete sampling of scale space, explicitly computing $I_\sigma(x,y)$ for a small number of values $\sigma = \sigma_0 \ldots \sigma_{k-1}$; we then compute ridges for each of these smoothed images to capture structures of different sizes. The values of σ we use are: $\sigma_i = \sqrt{2}^{\,i}$ with level $i = 0, \ldots \log_2(\min(\mathrm{w},\mathrm{h}))$ where w, h are image width and height. These computations are carried out in a very efficient way using recursive filters [10]. In practice, if we know the dimensions of characters and text strings, values of i can be limited to a small range. For example in our database, scales 2 to 8 are sufficient.

Figure 2 shows several images and ridges detected at two scales on regions extracted from the image. We can see that for each text, one ridge corresponding to the center line of the text and several small ridges corresponding to the skeletons of characters have been detected. The structure "one center line and lots of small skeletons " is present for many kinds of text (scene text or artificial text) with different character sets (latin alphabet or ideograms). Figure 3 illustrates the independence on orientation of the ridge based text representation.

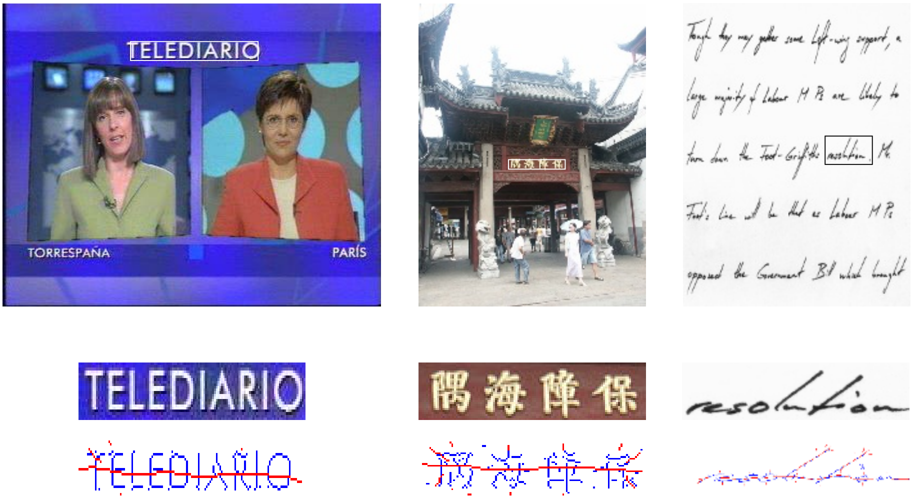
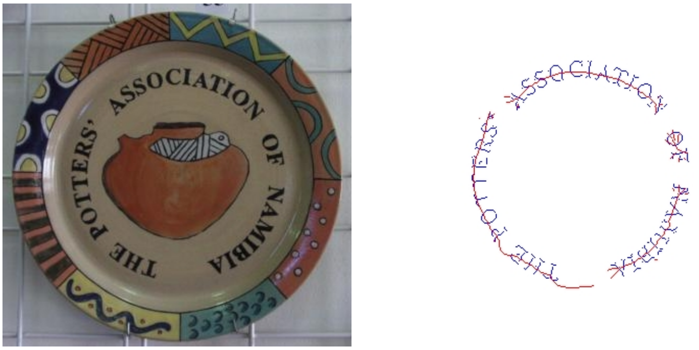


Fig. 2. First line: Images with rectangle showing the text region. Second line: Zoom on text regions. Third line: ridges detected at two scales (red in high level, blue in small level) in the text region that represent local structures of text lines whatever the type of text (handwritten text or machine text, scene text or artificial text, latin alphabet or ideograms).



(a)

(b)

Fig. 3. (a) Image of a plate. (b) Ridges detected at scale $\sigma = 2$ (blue lines) and $\sigma = 8\sqrt{2}$ (red lines). This figure shows the independence on orientation of the ridge-based text representation.

4.2 Classification of Candidate Text Blocks

The output of the previous step is k images containing ridge lines detected at k scale levels. Now, for each ridge at level i, $i = 0 \ldots k-1$, we classify the region corresponding to the ridge as text region or non-text region. The region corresponding to a ridge detected at scale σ is defined as the set of points such that the distance from each point to the ridge is smaller than σ. We call the ridge to be considered the *central ridge*, the region corresponding to the ridge the *ridge region* and all ridges at smaller scale in the ridge region which best fit character skeletons the *skeleton ridges*. The scale of the skeleton ridges is half the width of their strokes. It is not necessary that the skeletons and the center line be of the same type ("ridge"[1], "valley"[2]). We propose the following criteria to classify a region corresponding to a central ridge as text. Note that all detected ridges may be considered as central ridge starting from the largest scale σ_{k-1}.

- **Ridge Length Constraint**: Generally, the length of skeleton ridges representing the skeleton of the characters is approximately equal to the height of characters, which is 2 times the scale σ of the *central ridge*. For round characters like O, U, the length can reach up to 4 times σ. So the *skeleton ridge* length must be inside the interval $[\sigma, 4\sigma]$.
  Concerning the *central ridge*, supposing that $nbCharacters$ is the minimal number of characters in each text string, $minlength_{wc}$ is the minimal width of a character. Thus the length of the *central ridge* has to be longer than $nbCharacters * minlength_{wc}$.
- **Spatial Constraint**: With printed latin characters, *skeleton ridges* often are perpendicular to the *central ridge* at their center points. A text detector

[1] Local maximum of Laplacian.
[2] Local minimum of Laplacian.

should take into account this property. However, this is not true for some fonts (e.g. italic), and for other character sets (e.g. chinese or japanese). To construct a generic text detection system, we weaken the perpendicularity constraint by applying a non-parallel constraint. Thus, a text ridge region must contain an *important number of skeleton ridges* which are not parallel to the central ridge. Above, we supposed that there is at least $nbCharacters$ in the text string, as each character contributes at least one skeleton ridge, so the *number of skeleton ridges* inside the central ridge region has to be bigger than $\max\{nbCharacters, length_{centralridge}/minlength_{wc}\}$.

5 Experimental Results

5.1 Databases for Experiments

The databases for the testing algorithm contain single images and video frames. The first database (DB1) contains 10 images of a slide presentation. These images are taken by a camera with a resolution of 640x480 with various lighting conditions. The second database (DB2) consists of 45 images from news video[3], some of them having very complex background. The third database (DB3) contains 20 images extracted from formula 1 racing video[4] with a resolution of 352x288. Text in these images have different orientations (not limited to horizontal and vertical orientation) and undergo affine distortions. The fourth (DB4) contains 20 frames of film titles [5]. In this database, images contain text of different kinds (scene text, artificial text, and targeted scene text), sizes and styles. Table1 summarize these databases.

Table 1. Text detection result

	#images	#words	#detected words	#False alarms	Recall(%)	Precision(%)
DB1	10	172	172	7	100	96.09
DB2	26	103	99	48	96.11	66.67
DB3	45	217	169	114	77.88	59.71
DB4	20	199	177	18	88.9447	90.7692

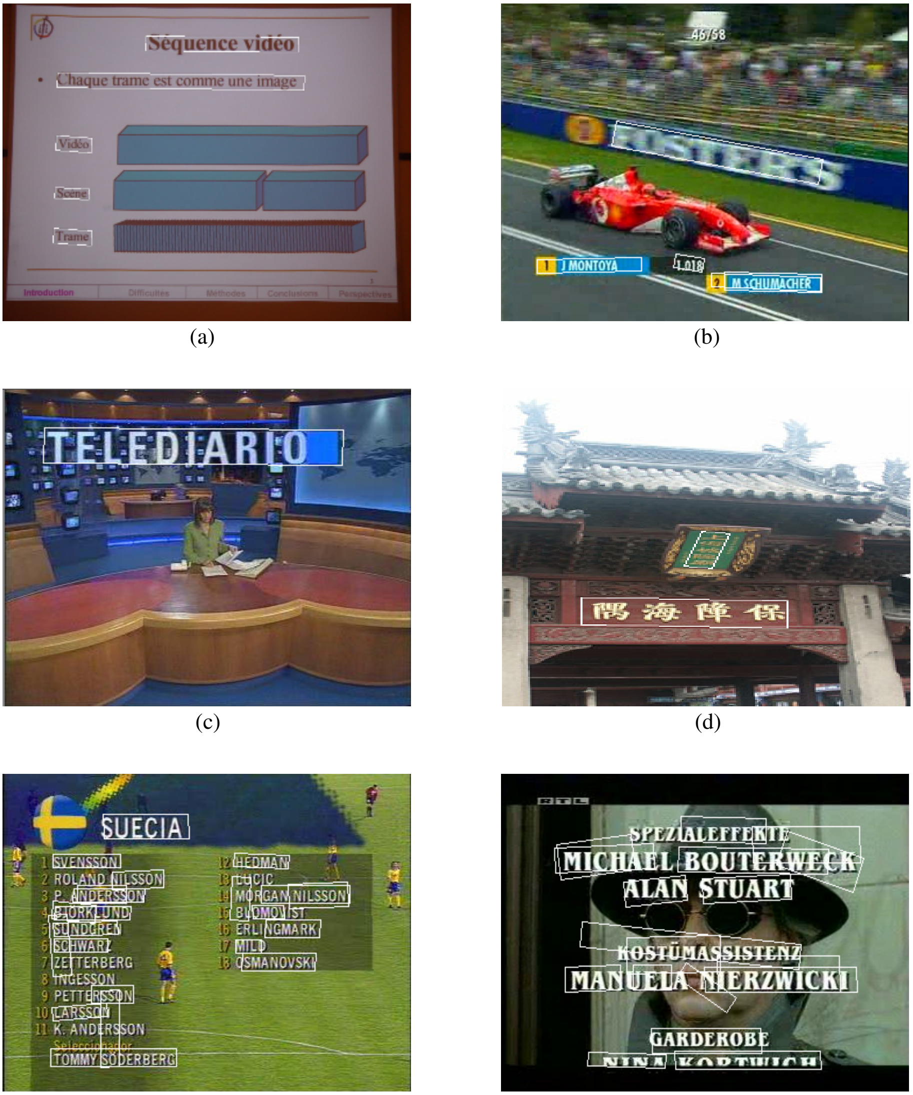
5.2 Evaluation

In our experiments, text size (the height of characters in the text in pixel) varies in the interval [4, 73], the text detection algorithm is computed only at $\lceil 2\log_2(73/2)\rceil = 11$ levels (while the maximal level is $N = \log_2(640x480) = 18$ with image of resolution of 640x480). The reason is that at scales coarser than

[3] $http : //www.cs.cityu.edu.hk/\ liuwy/PE\_VTDetect/$ used in [13] for evaluation of text detection

[4] $http : //www.detect - tv.com$

[5] $http : //www.informatik.uni - mannheim.de/pi4/lib/projects/MoCA$



(a) (b) (c) (d) (e) (f)

Fig. 4. Sample results of text detection. (a,b,c,d) When the background is homogeneous, detection is correct and does not give false alarms. (e) A table with text, without clear line structure (f) irregular background : there are false positives, and pieces of text are missed.

11, detected ridges represent structures of width larger than $2 * \sqrt{2}^{11} = 90$ pixels which are not textual structures, so ridges have no sense in the context of text detection. In fact, the number of levels to be considered can be determined based on prior information about the maximal size of text in image. In case where any information is provided, we use $N = \log_2(\mathrm{wxh})$, with w, h the width and the height of the image.

The minimum number of characters *nbCharacter* in each text string used is equal to 2 which appears reasonable because that we attempt to detect text lines, not isolated character. Moreover, we did not take into account points having the normalised Laplacian magnitude smaller than a threshold (here we used 5.0) in order to avoid false detections due to noise or aliasing artifact. As we have no information about the width of character stroke, we do not know exactly what is the scale of skeletons ridges. Thus for each central ridge detected at level k, the skeleton ridges at one among 3 levels $k-3$, $k-2$ and $k-4$ are taken as input of text constraints verifier. The choice of these 3 levels is based on hypothesis that the ratio of height and width of character stroke is in the interval [2, 4].

For evaluation, we use recall and precision measures. Table 1 shows the result of text detection from images in the 4 databases listed above. In the case of slides, we obtain the best recall as well as the best precision (figure 4a). All text regions in slide images are detected and localized correctly. The reason is that the background of slide image is well uniform and characters are distinctive from background. The detection was easily performed. With scene text having an orientation like those in images from the second database (Formula 1 car racing), the proposed algorithm had no difficulty (figure 4b). It is also well robust to noise and aliasing artifacts and it performs the detection of scene text as well as embedded text. In figure 4b, the score was not considered as a text because it appears too opaque in the scene. The performance of detection diminishes when the background is complex (images in the news video frame database) where there are cases of missed pieces of text and false alarms (figure 4e,f). The principal reason of false alarms is that the criterion “one center line and numerous small skeletons” also is satisfied by regions with regular grids. We either have to restrict our model, or these false responses have to be eliminated by an OCR system.

To compare with texture based and contour based methods, we implemented the texture based segmentation algorithm proposed in [12]. We found that with images in our databases, the clustering did not help to focus interest regions to be considered in a later stage. The contour based method fails in case where text is too blurred and scattered. In comparison with [12] where regions must be fused between scales because of “scale-redundant” regions, our approach verifies regions at the largest scale first; if it is a text region, this region will be no more considered later on. Without scale integration, the computation time is reduced significantly.

6 Conclusion

In this paper, we have proposed a novel approach for text analysis and text detection. Unlike traditional approaches based mainly on edge detection and texture, we use ridges as characteristics representing the structure of text lines at different scales. The experimental results show good recall and precision of the method using ridges (average of 90.7% and 78.3% respectively). The strengths of the method lie in its invariance to the size and the orientation of characters, its invariance to the form and the orientation of the lines, and that it works without any change in parameters for different writing systems (alphabets, ideograms). In addition, based on the scales at which we detect the central ridges and the skeleton ridges, the height, the number of characters in the text lines are measured. The current method still gives some false alarms, that can be eliminated by adding constraints on color and length between characters in text string or by using an OCR system.

References

1. D. Chen and J.M. Odobez J.P. Thiran. A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. *Signal Processing: Image Communication*, (19), 205-217 2004.
2. P. Clark and M. Mirmehdi. Combining statistical measures to find image text regions. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 450–453. IEEE Computer Society, September 2000.
3. P. Clark and M. Mirmehdi. Finding text regions using localised measures. In *Proceedings of the 11th British Machine Vision Conference*, pages 675–684. BMVA Press, September 2000.
4. David Eberly. *Ridges in Image and Data Analysis*. Kluwer Academic Publicshers, 1996.
5. H. Li and D. Doermann. A video text detection system based on automated training. In *Proceedings of the International Conference on Pattern Recognition ICPR'00*, 2000.
6. R. Lienhart. Automatic text recognition in digital videos. In *SPIE, Image and Video Processing IV*, pages 2666–2675, 1996.
7. A. Wernicke R. Lienhart. Localizing and segmentating text in images and videos. *IEEE Trans. Pattern Anal. Mach. Intell*, 18(8):256–268, 2002.
8. K. Sobottka and H. Bunke. Identification of text on colored book and journal covers. In *International Conference on Document Analysis and Recognition*, pages 57–62, Bangalore, India, September 1999.
9. H. Tran and A. Lux. A method for ridge extraction. In *Proceedings of the 6th Asean conference on Computer Vision, ACCV'04*, pages 960–966, Jeju, Korea, Feb 2004.
10. L.J. van Vliet, I.T. Young, and P.W. Verbeek. Recursive gaussian derivative filters. In *ICPR*, pages 509–514, August 1998.
11. V. Wu, R. Manmatha, and E.M.Riseman. Finding text in images. In *Proceedings of the ACM International Conference on Digital Libraries*, pages 23–26, 1997.
12. V. Wu, R. Manmatha, and E.M. Riseman. Textfind: An automatic system to detect and recognize text in image. *IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI*, Vol. 21(No. 11):1224–1229, November 1999.
13. L. Wenyin X.S. Hua and H.J. Zhang. Automatic performance evaluation for video text detection. In *International Conference on Document Analysis and Recognition (ICDAR 2001)*, pages 545–550, Seattle, Washington, USA, September 2001.
14. A. K. Jain Y. Zhong, K. Karu. Locating text in complex color image. *Pattern Recognition*, pages 1523–1536, 1995.