Scene Text Extraction in Complex Images

Hye-Ran Byun[1], Myung-Cheol Roh[2], Kil-Cheon Kim[1], Yeong-Woo Choi[3] and Seong-Whan Lee[2]

[1]Dept. of Computer Science, Yonsei University, Seoul, Korea
{hrbyun, kimkch}@cs.yonsei.ac.kr
[2]Dept. of Computer Science and Engineering, Korea University, Seoul, Korea
{mcroh, swlee}@image.korea.ac.kr
[3]Dept. of Computer Science, Sookmyung Woman's University, Seoul, Korea
ywchoi@sookmyung.ac.kr

Abstract. Text extraction and recognition from still and moving images have many important applications. But, when a source image is an ordinary natural scene, text extraction becomes very complicated and difficult. In this paper, we suggest text extraction methods based on color and gray information. The method using the color image is processed by color reduction, color clustering, and text region extraction and verifications. The method using the gray-level image is processed by edge detection, long line removal, repetitive run-length smearing (RLS), and text region extraction and verifications. Combining two approaches improves the extraction accuracies both in simple and in complex images. Also, estimating skew and perspective of the extracted text regions are considered.

1 Introduction

Texts in natural scene images often contain the important summarized information about the scene. If we could find these items accurately in real time, we can design the vision systems for assisting navigation of the moving robots or the blinds. Much of the previous research has focused on the text extraction with the gray-level images. Zhong *et al.* [1] used a spatial variance by assuming that it is lower in the background regions than in the text regions. But, the ascending and descending characters are not well detected on their test images. Ohya *et al.* [2] presented a method with several restrictions on the texts such as they should be upright without slant or skew, distinctive to the background regions, and uniform in their gray values. After a local thresholding, a relaxation algorithm is used for the component merging. Several kinds of the natural images were tested, and the extraction results are dependent on the text slant or tilt. Lienhart *et al.* [3] extracted the text regions using block matching after splitting and merging the connected components. Their method also has restrictions on size, gray values, and directions of the texts, and it also has difficulties in finding texts with skewed or with several colors on the same texts.

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 329–340, 2002.

In recent years there have been several approaches for extracting the text regions on color images. Haralick *et al*. [4] proposed a method using a differential top-hats morphological operator. The method shows robustness to the light changes. H. K. Kim [5] presented a method based on color segmentation and color clustering with video frames, and the characters are assumed to lying horizontally with similar colors and size. But, the method used too many ad-hoc thresholds. Jain *et al*. [6] presented two different methods and combined the results with complex color images. The first method used color quantization and connected components analysis on each quantized color plane. The second method used a spatial variance on the converted gray-level image. P. K. Kim [7] presented a method for complex color images. By using local quantization of the colors, text regions mixed with the backgrounds are removed. This method improved the extraction accuracy, but it requires plenty of computation time.

In this paper we extract text regions of natural scene images with two different methods, one in color image and the other in gray-level image. Then, we combine the two methods. Compared to the previous approaches to solve the same or similar problems, our approach has the following distinctive features: 1) To improve the color clustering results a new method is used in RGB color space; 2) To emphasize only the text regions accurately, line components surrounding text regions are removed and repetitive RLS is applied on the gray-level image; and 3) Estimating skew and perspective of the extracted text region is proposed.

2 Text Extractions in Color Images

The extraction method in color images consists of three steps: preprocessing, color clustering, and text region extraction and verification. We assume the colors of the characters in the same text regions are similar in this paper.

2.1 Preprocessing

The preprocessing consists of geometrical clustering, color reduction, and noise reduction. The input image size is 320x240, and the depth of each pixel is 24 bits. To reduce computation time the color reduction is needed and performed by dropping several lower bits of each pixel. During this process, the similar colors that we want to be clustered into the same color are often clustered into different colors. Thus, we first use a geometrical clustering. The geometrical clustering fills the pixels horizontally between the vertical edges with the same color. The vertical edges are found using equations (1) to (3) with a 3x3 mask as shown in figure 1. Equation (1) defines the strength of the vertical edge based on the Euclidean distance, *D*, defined by (2). The v_0, ..., v_8 represents the pixels, and v_R , v_G , v_B are the color components of each pixel. Figure 2 compares the results without and with the geometrical clustering. The edges produced by the proposed method are more accurate and clear. To save the computation, only the filling between the vertical edges is used.

$$E(v_0) = D(v_1, v_3) + 2 * D(v_4, v_5) + D(v_6, v_8) \quad (1)$$

$$D(v_1, v_2) = \sqrt{(v_{1_R} - v_{2_R})^2 + (v_{1_G} - v_{2_G})^2 + (v_{1_B} - v_{2_B})^2} \quad (2)$$

$$v = (v_R, v_G, v_B) \quad (3)$$

v_1	v_2	v_3
v_4	v_0	v_5
v_6	v_7	v_8

Fig. 1. Edge detection mask for the geometrical clustering

Fig. 2. Edge detection results: a given image (left), edge detection without (middle), and with (right) the geometrical clustering

The color reduction is realized by dropping the lower six bits of each RGB component. Thus, the reduced color image can represent up to 64 colors. During this process, a connected component like a stroke in the characters frequently gets disconnected due to noise. A 3x3 mask is used to eliminate the noise and to improve the connectivity of the strokes. The center pixel of the mask is compared with the neighboring pixels. If the number of pixels that have the same color with the center pixel is less than a given threshold, the center pixel is considered as a noise, and is substituted by the majority color in the mask.

2.2 Color Clustering

Since the image contents are sensitive to shading and surface reflections, the pixels in the same text components can be clustered into different colors during the color reduction. Also, since using all the color planes resulting from the color reduction requires a heavy computational cost, the color clustering is performed to reduce the number of color planes.

For 6-bit color clustering, a color histogram is first calculated. Then, the first color for the clustering is selected from the colors that are located at the corner points in the RGB color space with a largest color histogram. If there is no color at the corner points, then the colors located at the nearest to the corner points are the candidates. The nearest color to this beginning color is found by measuring the Euclidean distance between the two colors. If the distance is 1, the two colors are merged into the color with a larger histogram. The next color for clustering is selected by finding the longest distance from the previous merged color. This process is continued until the number of remaining colors is less than a given value, 3 in this paper, or until no more colors are merged. In this way, two complementary or near complementary colors that are usu-

ally texts and backgrounds in the natural scene images are rarely merged. Figure 3 shows a result of the color clustering: 30 colors remain after the color reduction, but only 6 colors remain by applying the proposed clustering method. Figure 4 shows the image of the color clustering. Thus, the image can be separated into 6 color planes. For each color plane, a morphological closing with 3x3 structuring element is applied to improve the connectivity of the strokes in the characters.

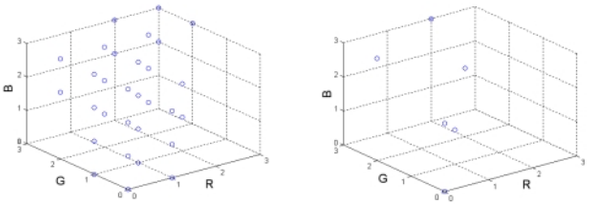


Fig. 3. Results of color reduction (left) and color clustering (right) in RGB color space

Fig. 4. Original image (left) and its clustering result (right)

2.3 Extracting Text Candidates

For each color plane we determine connected components and its size, bounding box of each component and its size, location and aspect ratio. Figure 5 shows the connected components and their bounding boxes. We only show four planes out of six color planes in the above clustering results.

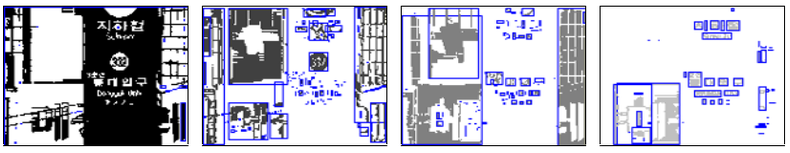


Fig. 5. Bounding boxes of the connected components for each color plane

Then, the connected component is removed when its size is too big or its width or height of the bounding box is too large or too small. The bounding boxes with its height smaller than 7 or larger than 90 pixels are removed. Figure 6 shows the results for each color plane and we can see many of the non-text components are removed.

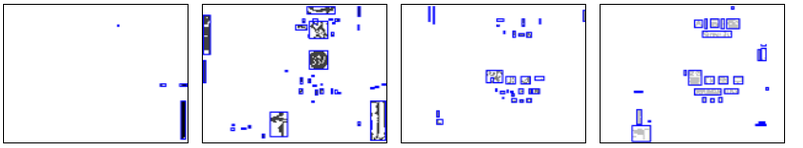


Fig. 6. Removal of non-text components by considering component size, width or height

Next, the remaining boxes are merged to make characters or text lines. Closeness and overlapping ratios between the bounding boxes are examined for merging, and its results are shown in figure 7.

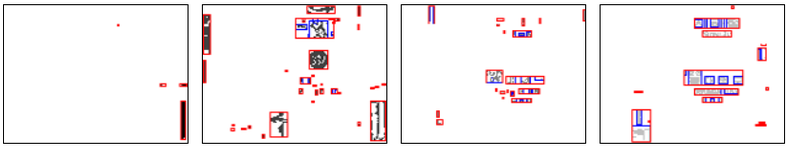


Fig. 7. Merging bounding boxes to make characters or text regions

We again remove some of the bounding boxes by considering the pixel density of the bounding box. The pixel density of the candidate text region is usually larger than that of the non-text bounding box. Figure 8 shows the results.

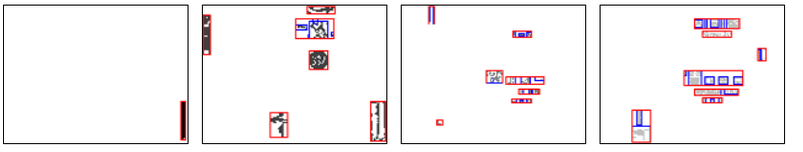


Fig. 8. Removal of the merged boxes as non-texts

Since the text regions usually contain densely packed edges compared to other regions, some of the bounding boxes are further removed by considering the edge distribution. As an example, a region on a wall painted with the same color due to noise or light influence can be easily removed. Figure 9 shows the result. Only three planes maintain the bounding boxes of the candidate text regions with this removal.

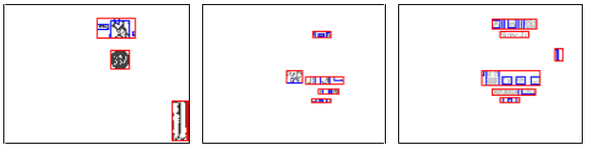


Fig. 9. After verifying each box with edge distribution

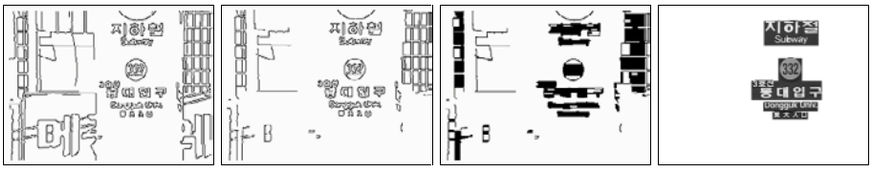
Finally, considering the overlapping ratio of the bounding boxes combines the results from each color plane. Figure 10 shows the combined result, before and after.

Fig. 10. Bounding boxes obtained in each color plane (left) and their combined results (right)

3 Text Extractions in Gray-Level Images

Since the text regions in the natural scene images have edges densely populated and they are usually surrounded rectangular boxes with the relatively long lines, our approach finds the text regions based on the edge density, and removes the long line elements that can border the text regions during the region emphasis. Also, the long line elements around the text regions are used for the skew and perspective estimations. There are four steps in this approach: preprocessing, text region extraction, verification, and skew/perspective estimations.

A median filter is applied to the 320x240 gray-level image for the preprocessing, then the edges are found using the Canny edge detection method, as shown in figure 11(a).



(a) (b) (c) (d)

Fig. 11. Images of (a) edges, (b) long lines removed, (c) RLS applied, (d) extraction results

3.1 Extracting Text Regions

The long line components are first found and removed in the edge image. The long line components include horizontal or vertical lines, quadrilaterals with long lines, and broken lines with long horizontal or vertical components. The 8-directional edge following is performed and the histograms of each direction are obtained. Horizontal or vertical longs lines have the dominant histograms only in one direction. Quadrilaterals have dominant histograms usually in two directions. And, the broken lines also have the dominant histograms in two or three directions. Since the 8-directional histogram

bins can be different according to the starting point of the edge following, we also consider 4-directional bins by adding the opposite direction bins together. Then the long line elements will become the connected components with the dominant values of the histograms in either one or two directions both in 8 and 4-directional bins. Figure 11(b) shows the result with the long lines removed.

Then, repetitive RLS is applied to emphasize the text regions, since the edge density of the text regions is usually high. The RLS is applied iteratively since there are various distances between text edges due to size variations in fonts and various inter-distances between character strokes. By increasing the run length from small to large horizontally and vertically the text regions are emphasized more accurately. The results are shown in figure 11(c).

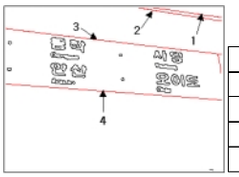
3.2 Verifying Candidate Regions

After applying repetitive RLS, we remove the connected components by analyzing the components and their bounding boxes. First, component with a large number of pixels is removed. Those components, for example, can be leaves on a tree or tiles on a wall with dense edges. Also, component with a small number of pixels is removed, since even though the regions are texts they can be hardly recognized due to their small size. Then, we further removed the components when its size is too small comparing to the size of its bounding box. Also, if there are components that have a large aspect ratio or whose widths or heights are less than 5 pixels, they are removed. The results are shown in figure 11(d).

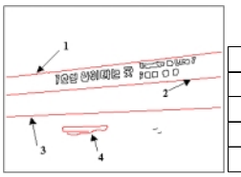
3.3 Skew and Perspective Estimations

After the text regions are extracted, the skew and perspective of the regions are estimated. The long lines surrounding or around the text regions are used to estimate the skew angle. By using a least mean square method to each line component, slope, location, and mean square errors are found.

For the skew estimation, we first apply a simple smoothing to the line elements to remove small bumps, holes, and disconnections. Since we are currently focusing on extracting the horizontal text lines, only the lines with small slopes are smoothed. For each smoothed line element 10 points are sampled for the slope and location estimations. The sample points are obtained by dividing the line components by 10 equal distances. When there exists a line that is almost straight but with small abrupt branches as shown in line 3 of figure 12(a), the slope of this line can be found by sampling 10 points only from the component region that has the maximum histogram in the 4-directional bins. The 10 sampling coordinates are applied to the least mean square method. During this process a component with either a curve or with many branches can be discarded by removing a line with a large mean square error. The line 4 in figure 12(b) is removed due to its high error value.



No.	Slope	Y-intercept	Error
1	-0.14	268.8	0.65
2	-0.15	267.1	1.02
3	-0.11	206.7	0.48
4	-0.07	127.5	0.82



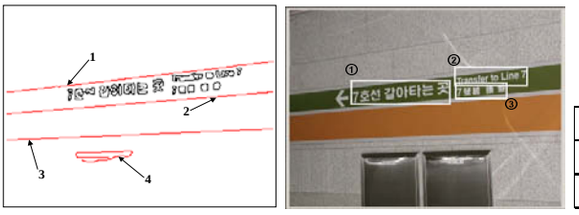
No.	Slope	Y-intercept	Error
1	0.11	135.9	0.94
2	0.08	110.7	0.63
3	0.04	79.6	0.68
4	-0.01	178.6	197.95

(a) (b)

Fig. 12. Examples of finding slopes, locations, and mean square errors

For an accurate estimation of the skew angle the lines are merged when they are located in the vicinity of the same text region. For the lines with small mean square errors we consider both slope and location (*Y*-intercept) for merging. First, the slopes of the two lines are compared, and when they are very similar, two lines are candidates for merging. Then, the locations are compared, and when they are in close proximity, they are merged. The merged line has slope and location though averaging the two candidate lines. In figure 12(a), lines 1 and 2 are merged, but lines 3 and 4 are not.

When there are two lines surrounding a center of the text box, the skew angle and perspective can be measured. If the slopes of the two lines are very similar, we regard the text region has only skew and the skew angle is used for the correction. But, when the slopes are not similar, there is a perspective. The perspective can be measured by finding four intersection points with two vertical end lines of the image and by subtracting the two vertical distances. When there is only one line located near a text box, only the skew angle can be estimated. The skew and perspective of the text regions are estimated from the lines 1 and 2 of figure 13. The line 3 is not used. The skewed images are corrected by applying shearing transformations as shown in figure 14.



No.	①	②	③
Skew	6.65	6.65	4.74
Perspective	9.57	9.57	9.57

Fig. 13. Examples of skew and perspective estimations

Fig. 14. Skew image (left) and its corrected result (right)

4 Combining Two Approaches

The color-based method is sensitive to the lighting conditions, and the gray-based method is sensitive to the complexity of the background. The former is rather robust to the background complexity, and the latter robust to the lighting conditions. Thus, combining two methods can complement the shortcomings of each method. When the extracted text regions in each method have similar locations and sizes, we conclude the regions as the correct ones and the verification by the other method is skipped. But, when a region is detected as the text region only in one method, this region is verified by the other method. Verifying this local region only can reduce the effect of surrounding edges/objects in the gray-based method, and can expect the improvement of the color clustering in the color-based method. Combining the two methods is shown in figure 15.

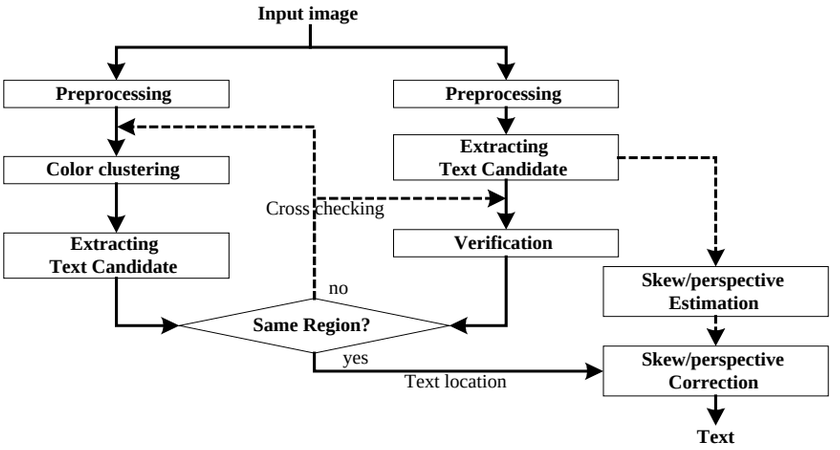


Fig. 15. Combining two methods

When a candidate text region is only detected in the color-based method, that local region is verified. Figure 16(a) shows the regions that are extracted on the color-based method and figure 16(b) shows the corresponding regions with the repetitive RLS is applied on the gray-based method. Since the density of the connected component and the ratio of the bounding boxes in figure 16(b) are not similar to those of the text, these regions are rejected.

Fig. 16. False accepted regions with color-based method (left) and its corresponding regions after repetitive RLS applied (right)

When a text region is detected only in the gray-based method, the verification is also applied to that local region. Figure 17(a) shows the candidate text regions found in the gray-based method, and figure 17(b) shows their color clustering results in the color-based method. The color clustering results of these local regions are better than the result obtained by color clustering on the whole image. This is because the number of colors remaining after the color clustering is restricted to only two or three. But, for the whole image, the number of colors remained after the clustering can be larger than two or three. Thus, the regions in figure 17 are verified as non-texts by color-based verification.

Fig. 17. False accepted regions in gray-based method (left) and its corresponding regions after color clustering (right)

5 Experiments and Results

We have tested the proposed methods with 120 natural scene images. The images were captured in various places such as in schools, hospitals, subway stations, and streets. We then classified the images into two categories, simple and complex. The guidelines for the classifications are properness of text size, distinctiveness between texts and backgrounds, amount of skew/perspective, *etc*. The reason for dividing the images is to evaluate the proposed methods in different levels of difficulties.

Experimental results are summarized in Table 1. *Correct* is a number of text regions that are correctly extracted. *Missing* is a number of texts failed to detect. *False* counts the number incorrectly identified as texts. When the extracted text region does not contain whole text region, the region is counted as *correct* when the size of the extracted text region is more than two-third of the correct regions. Otherwise, it is counted as *false*.

Table 1. Extraction results with simple and complex test images

	Total number	Method	*Correct*	*False*	*Missing*
Simple images	255	Combining	202	124	53
		Color-based	194	156	61
		Gray-based	210	151	45
Complex images	235	Combining	177	207	58
		Color-based	190	282	45
		Gray-based	165	201	70

From Table 1, color-based method demonstrates better results than the gray-based method for complex images, but it has more false detections. The gray-based method has better performance for simple images. The color-based method gives steady per-

formance on both simple and complex images. Also, the results of the combining method are better than that of each method. The number of false detected regions is significantly reduced, and the number of missing is also reduced.

Figure 18 shows the extraction results. The results show that the color-based method continues to make errors where surface reflection is strong, while the gray-based method fails when the background is complex. In the bottom images of figure 18, the first image is a combined result that does not require cross verifications. The second underwent the cross verifications since some of the bounding-boxes do not coincide in the two results. We can see the usefulness of the cross validations at the bottom images in figure 18.

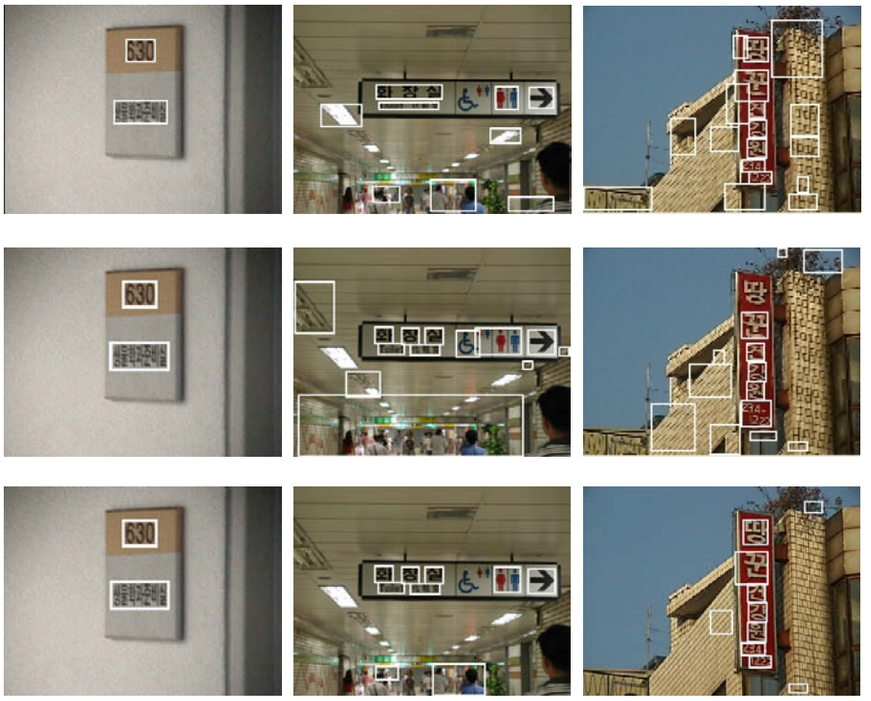


Fig. 18. Extraction results with color-based method (top), with gray-based method (middle), and combined results (bottom)

6 Conclusions

We attempted to extract texts from natural scene images with only a few restrictions. We suggested two different methods for the text extraction, and a method for estimating skew and perspective of the candidate text regions was proposed. Our approach has the following distinctive features: 1) To improve the color clustering results a new method is used in RGB color space; 2) To emphasize only the text regions accurately,

line components surrounding text regions are removed and repetitive RLS is applied on the gray-level image; and 3) Estimating skew and perspective of the extracted text region is proposed. With these features our approach shows the improved extraction accuracies both in simple and in complex images in the experiments.

Acknowledgements. This research was supported as a Brain Neuroinformatics Research Program sponsored by Korean Ministry of Science and Technology (M1-0107-00-0009).

References

[1] Y. Zhong, K. Karu and A. K. Jain, "Locating Text in Complex Images," Pattern Recognition, Vol. 28, No. 10, pp. 1523–1535, 1995.
[2] J. Ohya, A. Shio and S. Akamatsu, "Recognizing characters in scene images," IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-16(2), pp. 67–82, 1995.
[3] R. Lienhart and F. Stuber, "Automatic text recognition in digital videos," Image and Video Processing IV, SPIE, 1996.
[4] L. Gu, N. Tanaka and R. M. Haralick, "Robust Extraction of Characters from Color Scene Image using Mathematical Morphology," Proceedings of International Conference on Pattern Recognition, Vol. 2, pp. 1002–1004, 1998.
[5] H. K. Kim, “Efficient automatic text location method and content-based indexing and structuring of video database,” Journal of Visual Communications and Image Representation, Vol. 7, pp. 336–344, 1996.
[6] A. K. Jain and Bin. Yu, "Automatic text location in images and video frames," Pattern Recognition, Vol. 31, No. 12, pp. 2055–2076, 1998.
[7] P. K. Kim, "Automatic Text Location in Complex Color Images using Local Color Quantization," Proceedings of IEEE Region 10 Conference, Vol. 1, pp. 629–632, 1999.