PAPER

A Contour-Based Robust Algorithm for Text Detection in Color Images

Yangxing LIU[†a)], ***Student Member***, **Satoshi GOTO**[†b)], ***Fellow***, ***and*** **Takeshi IKENAGA**[†c)], ***Member***

SUMMARY Text detection in color images has become an active research area in the past few decades. In this paper, we present a novel approach to accurately detect text in color images possibly with a complex background. The proposed algorithm is based on the combination of connected component and texture feature analysis of unknown text region contours. First, we utilize an elaborate color image edge detection algorithm to extract all possible text edge pixels. Connected component analysis is performed on these edge pixels to detect the external contour and possible internal contours of potential text regions. The gradient and geometrical characteristics of each region contour are carefully examined to construct candidate text regions and classify part non-text regions. Then each candidate text region is verified with texture features derived from wavelet domain. Finally, the Expectation maximization algorithm is introduced to binarize each text region to prepare data for recognition. In contrast to previous approach, our algorithm combines both the efficiency of connected component based method and robustness of texture based analysis. Experimental results show that our proposed algorithm is robust in text detection with respect to different character size, orientation, color and language and can provide reliable text binarization result.

key words: *text detection, texture analysis, connected component analysis, region contour, edge detection*

1. Introduction

The retrieval of text information from color images has gained increasing attention in recent years. Text appearing in images can provide very useful semantic information and may be a good key to describe the image content. Text detection can be found in many applications, such as road sign detection, map interpretation and engineering drawings interpretation etc. Many papers about the text detection from color images or video sequence have been published. However, due to the complexity of text appearance in color images, text detection is still a difficult and challenging task in image processing.

Based on the ways being used to locate text regions, most of the text detection techniques can be classified as either connected component (CC) based or texture-based algorithms.

The first class [1]–[5] is based on the analysis of the geometrical arrangement of edges or homogeneous color and grayscale components that belong to characters. CC-based methods segment an image into a set of CCs, group small CCs into successively larger ones, and then classify the final CCs as either text or non-text by analyzing their geometrical characteristics. The CC based algorithms are relatively simple to implement, but they are not very robust for text localization in images with complex background. Chen et al [3] detected vertical and horizontal edges in an image and dilated two kinds of edges using different dilation operators. Real text regions are then identified by using support vector machine. Zhong et al [4] extracted text as those CCs that follow certain size constraints and horizontal alignment constraints. Jain and Yu [5] analyze CC of same color to select foreground pixel and adopt structure properties to identify text. But this method can extract only horizontal texts of large sizes.

Texture-based [6]–[10] methods are based on the fact that texts in images have distinct textural properties that can be used to discriminate them from the background or other non-text region. Those methods can be classified into three major categories, namely, statistical, structural and spectral. In general, the texture-based algorithms are more robust than the CC-based algorithms in dealing with complex background [8]. The high complexity of texture segmentation is the main drawback of this method. The algorithm proposed in [9] using texture features to extract text but failed in the case of small font characters. In [10], wavelet features from fix-size blocks of pixels and classify the feature vectors into text or non-text using neural networks. Because the neural network based classification is performed in the whole image, the detection system is not very efficient in terms of computation cost.

Shortcomings of many current methods include their inability to perform well in the case of variant text orientation, size, language and low resolution image, where characters may be touching. Some methods [6], [9], [11] assume that the text direction is horizontal or vertical and text font size is in limited range. Some proposed methods [11]–[13] fail to detect isolated characters because there is no contextual information and certain text parameters, such as base line, character width and height, can not be accumulated with statistical method in such case. Some methods [14] concentrate on detecting individual letters. The method proposed in [15] needs to know text rectangle region position in advance, which is not available in most practical cases. So these algorithms are very restrictive in the type of texts they can process and therefore restrict the application domain of the text detection algorithm unnecessarily.

Manuscript received March 30, 2005.
Manuscript revised September 4, 2005.

[†]The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kita kyushu-shi, 808–0135 Japan.
a) E-mail: lyx@ruri.waseda.jp
b) E-mail: goto@waseda.jp
c) E-mail: ikenaga@waseda.jp

DOI: 10.1093/ietisy/e89–d.3.1221

This paper presents a novel method to precisely extract character regions in color images to make a readable image for OCR (Optical Character Recognition). The method is targeted towards being robust with respect to diverse kinds of text appearances, including character font size, orientation, color and language.

In contrast to previous approaches, our algorithm is a hybrid approach, which combines CC based method and texture analysis method. First, with an elaborate edge detection algorithm, we extract the edge pixels of all possible text regions in color images. CCs are clustered by linking those edge pixels to construct contours of all possible text regions. The gradient and geometrical property of region contours is checked to generate candidate text regions efficiently. This is followed by the analysis of texture features derived from wavelet domain in order to separate non-text regions and verify true text regions. Eventually, the Expectation maximization (EM) algorithm is introduced to binarize each text region to prepare data for OCR.

Many existing text detection algorithms [3] are based on character string (two or more neighboring characters aligned in the same direction) detection, while our algorithm detects characters individually. So our algorithm can detect isolated characters. Furthermore, because OCR engines can only perform on the binarized image, where black text characters appear on a white background, with many previous text detection algorithms character string needs to be binarized, segmented into characters before recognition. However, with our algorithm, which is based on character detection, we just binarize the detected text regions and feed the result to OCR. So much effort related to character segmentation is also saved.

Since our algorithm analyzes text region contours directly and not the entire image, which are more stable than complete color images under varying illumination conditions or related camera parameters, it is robust to varying illumination conditions. Furthermore, instead of performing a global texture analysis on the whole image like some existing algorithms [9], we consider candidate text regions separately to remove false alarms (i.e. non-text regions) and perform the texture analysis on the region contour, which makes our algorithm more efficient and robust.

The rest of the paper is organized as follows. Section 2 describes the proposed algorithm in detail. Experimental results are explained in Sect. 3. Conclusion remarks are given in last section.

2. Text Detection Algorithm

The proposed algorithm is mainly based on the following properties of text:

1. Contour gradient
   The pixels representing text contour usually have a high contrast to their neighbor pixels. So we can select those text contour pixels with high gradient magnitude.
2. Structural information
   Contours of text regions usually have relatively large gradient direction variance range and pixel number.
3. Texture property
   Text appearances are distinguished by a characteristic spatial pattern on the character level. The writing direction of one character, periodically strong constrast fluctuations can be notified. So we analyze texture features of region contour to verify true text regions.

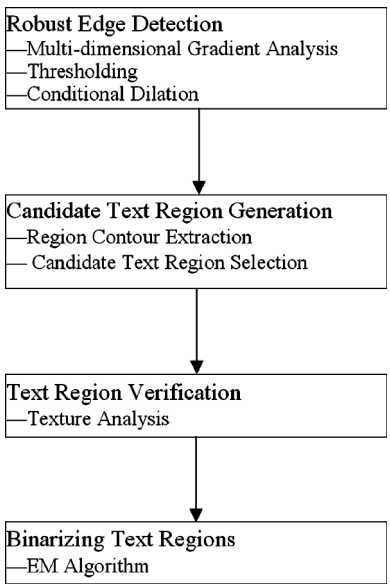


Fig. 1 Flow diagram of our algorithm.

The flow chart of our algorithm is shown in Fig. 1. In the following, we explain and state the details of four main steps in order of processing.

2.1 Edge Detection

Edge detection is an important pre-processing step of our algorithm, although edge detection is not the focus of this paper. With good edge detection result, the time complexity of later text detection will be reduced and the detection accuracy will be improved.

Accurate edge information leads to accurate geometrical information and accurate discrimination between text and non-text regions.

An important fact we observed is that the pixels representing text contour usually have a high contrast to their neighbor pixels. In our algorithm we use black pixels (represented by 1's) to represent the edge pixels and white pixels (represented by 0's) to represent non-edge pixels.

The main objective of this stage is to extract precise edge information of all text regions and filter out edges of most non-text regions.

Because the edge detector we adopted is different from most existing isotropic edge detector, which can not provide accurate edge direction information, basic ideas are first presented below to illustrate the edge detector. Some ideas about the detector have been described in [16].

2.1.1 Edge Detector

Given a color image, the difference vector DV in rgb color space induced by moving an infinitesimal step in the image plane in the direction dx,dy is : DV=(dx dy)J_c^T,

$$J_c = \begin{bmatrix} \partial r \partial x & \partial r/\partial y \\ \partial g/\partial x & \partial g/\partial y \\ \partial b/\partial x & \partial b/\partial y \end{bmatrix}$$

where J_c is the Jacobian matrix of the image, r_x, r_y, g_x, g_y, b_x and b_y, are the six first order derivatives of the three color channels with respect to the two coordinates x and y.

The Euclidean squared magnitude of DV is

$$DV^2 = (dx \quad dy) M_c (dx \quad dy)^T,$$
$$\text{where} M_c = J_c^T J_c = \begin{bmatrix} M_{xx} M_{xy} \\ M_{xy} M_{yy} \end{bmatrix},$$
$$M_{xx} = (r_x)^2 + (g_x)^2 + (b_x)^2$$
$$M_{xy} = r_x * r_y + g_x * g_y + b_x * b_y$$
$$M_{yy} = (r_y)^2 + (g_y)^2 + (b_y)^2$$

and DV^2 is a measure of the rate of change of the image in the direction of dx,dy. Maximizing this magnitude is an eigenvalue problem. We can obtain the magnitude extremum in the direction of the eigenvector of the matrix M_c and the extremum value is the corresponding eigenvalue. The trace of M_c, $((r_x)^2 + (g_x)^2 + (b_x)^2 + (r_y)^2 + (g_y)^2 + (b_y)^2)$ is evaluated as a measure of the joint channel gradient intensity.

The larger eigenvalue of M_c is

$$V = (\sqrt{(M_{xx} + M_{yy})^2 - 4 \times (M_{xx} \times M_{yy} - M_{yy}^2)} + M_{xx} + M_{yy})/2 \quad (1)$$

and its corresponding eigenvector is $\{M_{xy}, V - M_{xx}\}$. The gradient direction angle is defined by eigenvector:

$$\theta = \arctan\left(\frac{V - M_{xx}}{M_{xy}}\right) \quad (2)$$

The square root of the larger eigenvalue and its corresponding eigenvector direction are the equivalents of the gradient magnitude and gradient direction at any given point. So we can get precise gradient magnitude and direction of each pixel (i,j) by computing corresponding eigenvalue V(i,j) and eigenvector direction $\theta(i, j)$ with Eqs. (1) and (2).

2.1.2 Edge Extraction Algorithm

First, median filter was applied to the input image C to reduce the noise of the input images while preserving sharp edges.

Then the matrix M_c is computed for each pixel on the image and we will get a series of eigenvalues V and eigenvectors E. In Fig. 3, a grey level image, the grey level value

Fig. 2 Original color image.

Fig. 3 Illustration of gradient magnitude.

of each pixel (i,j) equals to the gradient magnitude (eigenvalue V(i,j)) of its corresponding pixel in Fig. 2

As we know, edge pixels are those points with local maximum gradient magnitude in their gradient direction. Furthermore, we also notice that the edges of text symbols are typically stronger than those of noise or background areas. As we can see from Fig. 3, texts have strong edge magnitude.

So a pixel (i,j) is accepted as a possible text contour pixel only if it meets the following two requirements.

First, the pixel must have a larger gradient magnitude than that of its neighbor located in the direction closest to its gradient direction, i.e., the eigenvalue of an edge pixel must be greater than that of both two neighboring pixels, which are closest to its eigenvector direction.

Second, the pixel gradient magnitude (i.e. eigenvalue V(i,j)) must be greater than the adaptive threshold T to eliminate weak edges. T is determined by the following formula:

$$T = \frac{\sum_{(i,j)\in C}(V(i,j) \times |V_{dif}(i,j)|)}{\sum_{(i,j)\in C}|V_{dif}(i,j)|} \quad (3)$$

$$V_{dif} = V(i', j') - V(i'', j'') \quad (4)$$

where C represents the color image, both pixel (i', j')

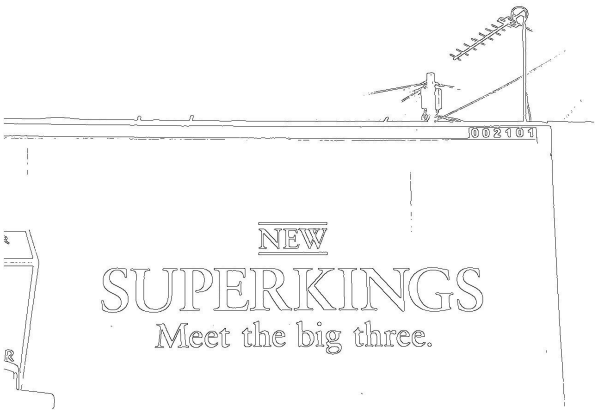


Fig. 4 Edge map of possible text regions in Fig. 2.

and (i'', j'') are 8-neighboring pixels and closest to eigenvector direction of pixel (i,j). V_{dif} is the corresponding eigenvalue difference between these two pixels.

To make character contour more continuous, conditional dilation is performed on edge collection obtained in previous operation to connect text edges to form closed contours. A 3×3 square structuring element with the origin at its center is selected to dilate the image edge. Furthermore, the gradient magnitude of center pixel (i_c, j_c) must exceed T and the gradient direction (corresponding eigenvector direction) difference between the center pixel and its neighboring edge pixel (i_n, j_n) must be less than a predefined angular tolerance α,which is based on the fact that that the gradient direction variance of two adjacent edge pixels in the same region is very small. The value of α has been experimentally set to 0.26, about 15 degrees. So only if inequality (5) is true, the pixel (ic, jc) will be dilated as an edge pixel.

After this, all character edge pixels as well as some non-character edge pixels which also show high local color contrast are remained in the image edge map. Then we can link connected edge pixels to form region contour through CC analysis to generate text region candidates.

As we can see from Fig. 4, edge pixels of all text regions in Fig. 2 (1280*960 pixels) are remained and most non-text region edge pixels are excluded.

$$V(i_c, j_c) > T \quad \& \quad |\theta(i_c, j_c) - \theta(i_n, j_n)| < \alpha \tag{5}$$

2.2 Generation of the Candidates of Text Regions

In this phrase, we first need to group connected edge pixels into different region contours through CC analysis.

A set of pixels in an image which are all connected to each other is called a CC. Unlike previous work [1], in which CCs are extracted by grouping neighboring connected pixels belonging to text regions, we extract CCs directly from region contours.

There are several ways to define the notion of the connectivity. In digital images, the definition of the CCs is the

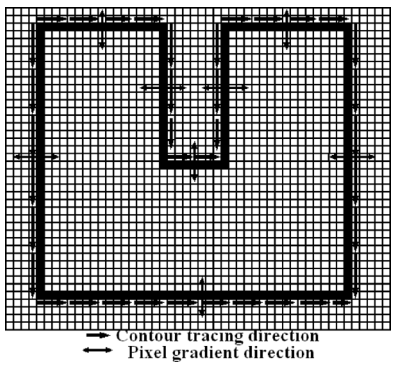


Fig. 5 Illustration of contour tracing process.

definition of a local neighborhood describing the connections between adjacent pixels. First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values SV. With a binary image, we simple have SV=1. Second, a path must exist between two connected pixels on which each pixel is 4(8)-connected to next one. Given a pixel, its 4(8)-connected pixels must be from its 4(8)-neighbor pixel set. Texts in images are formed out of CCs that in turn are made of connected pixels.

Our method is based on the principle that a character is fully determined by its contour, just as a polygon is fully determined by its vertices.

2.2.1 Region Contour Extraction

Our region contour tracing procedure basically follows the 8-neighborhood-connectivity algorithm discussed below.

Scan the image from left to right and from top to bottom. Initialize the class label of each edge pixel CL(i,j) to number 0. Given an edge pixel P, examine the four neighbors of P, which have already been encountered in the scan (i.e. the neighbors to the left of P, above it, and the two upper diagonal terms). Then we can label P according to following different cases.

- If only one neighbor has class number bigger than 0 and the gradient direction difference between this neighbor and P is less than α, assign this class number to P, else
- If more than one neighbor has class number bigger than 0 and the gradient direction difference among these neighbor pixels and P is less than α, assign one of the labels to P and make a note of the equivalences, else
- Assign a new class number to P.

After completing the scan of the whole image, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. Then a second scan is made through the whole image, during which each label is replaced by the label assigned to its equivalence classes. Figure 5 shows the contour tracing process of a closed region.

Applying the algorithm described above, we can link edge pixels of a possible character into a closed contour and

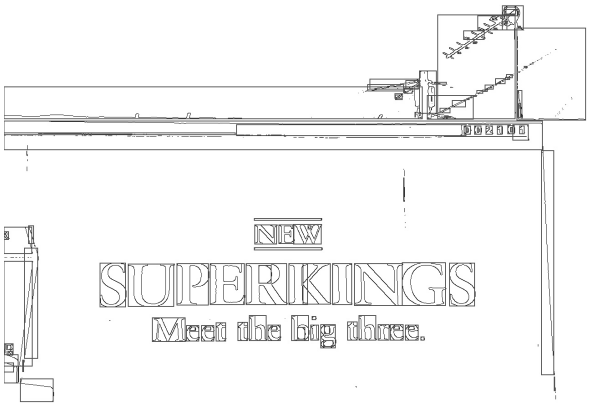


Fig. 6 Illustration of region contour extraction result.

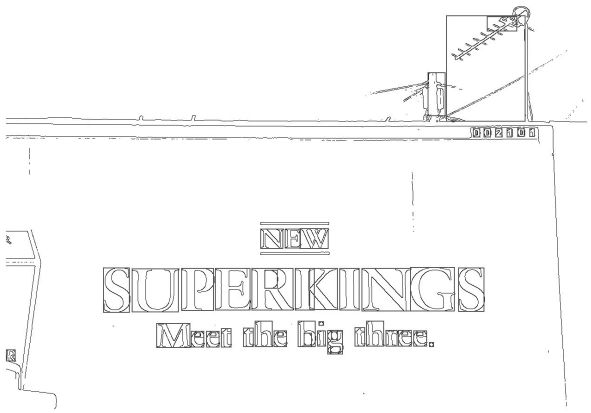


Fig. 7 Illustration of candidate text regions.

obtain many CCs from edge map. The next stage is to select part of the components as candidate text regions and remove false alarms. The leftmost, rightmost, topmost and bottommost coordinates of each component are taken to create the bounding rectangles. In Fig. 6, all region contours linked by CC analysis are enclosed by blue rectangles.

2.2.2 Selection of Candidate Text Regions

To develop the criteria for filtering out non-text regions, the following features of text are observed:

1) Average gradient magnitude of contour pixels

The first fact is that the average gradient magnitude of edge pixels is usually higher for text than non-text blocks. We can calculate this average value of each CC region with following formula:

$$V_{avg} = \frac{\sum_{(i,j)\in R} V(i,j)}{m}, (CL(i,j) = CL_R), \tag{6}$$

where m is the number of pixels labeled with one CC class CL_R representing the region R. V_{avg} of a text region should be greater than 2*T.

2) Gradient direction variance of contour pixels

Another important fact we noticed is that text regions have a higher gradient direction distribution variance than graphic regions. The variance of a region can be estimated by $\theta_{max} - \theta_{min}$, where θ_{max} and θ_{min} are the maximum and minimum edge gradient direction angle respectively in a CC region. Within a text region, the variance must be greater than PI(180 degree), i.e. $(\theta_{max} - \theta_{min}) > PI$.

3) Number of contour pixels

The third valuable fact we observed that text block should have more edge pixels than some non-text blocks. The edge pixel count in a text block, labeled as the same class within a CC region, must be greater than MAX(2 * W, 2 * H), where W and H are width and height of the CC region respectively.

With features described above, three criteria are applied on every CC in order to reject blocks constructed of noise pixels and classify some non-text blocks.

The output of this step is a rectangular box array, each of which circumscribes a candidate text region. Some text like region remains after CC analysis so we need to perform texture analysis to further distinguish text regions from non-text regions, which also have high edge density and strength.

In Fig. 7, regions enclosing by blue rectangular boxes are candidate text regions of Fig. 2. Most non-text region contours in Fig. 6 are eliminated.

2.3 Text Region Verification

The candidate text regions generated in previous section may include false alarms. Because text shows a rhythmic spatial pattern distribution, which consists of a regular alternation of contrast changes in one specific direction, texture analysis can be exploited to separate non-text regions like graphics, images and other non-text rectangular homogeneous and high contrast regions. Instead of performing a global texture analysis on the whole image, we consider every region of interest separately to remove false alarms, which makes our algorithm more efficient and robust.

Texture is a well-researched property of text regions. The discrete wavelet transform (DWT) can well catch texture feature of arbitrary regions [17], which is a very useful tool for signal analysis and image processing. So we utilize two features in wavelet domain, the second and third order wavelet moment, to classify some non-text regions.

We select the Harr wavelet as the basis for our texture characterization due to its two prominent merits:

- Good ability to characterize texture features.
  Haar wavelets are real, orthogonal, and symmetric.
- High computation efficiency [18].
  The high-pass filter and the low-pass filter coefficient are simple (either 1 or -1).

Let $\phi(x)$ and $\psi(x)$ be the two Haar wavelet bases of one-dimensional.

$$\begin{cases} \phi_{k,s}(x) = 2^{-k/2}\phi(2^{-k}x - s) \\ \psi_{k,t}(x) = 2^{-k/2}\psi(2^{-k}x - t) \end{cases}$$

With 2-dimensional image data, the corresponding tensor product transform basis can be calculated as follows:

$$\begin{cases} \varphi^{LL}_{k,s,t}(i,j) = \phi_{k,s}(i)\phi_{k,t}(j) \\ \varphi^{LH}_{k,s,t}(i,j) = \psi_{k,s}(i)\phi_{k,t}(j) \\ \varphi^{HL}_{k,s,t}(i,j) = \phi_{k,s}(i)\psi_{k,t}(j) \\ \varphi^{HH}_{k,s,t}(i,j) = \psi_{k,s}(i)\psi_{k,t}(j) \end{cases}$$

Because the input image is colored, first the input image is converted into grey-level image I using formula:

$I = 0.299 \times r + 0.587 \times g + 0.114 \times b$.

Then the image I is processed with discrete wavelet transform and transformed into four sub-bands LL, HL, LH and HH with a 2-channel filter bank (L: low pass filter, H: high pass filter). LH represents the horizontally low-frequent and vertically high-frequent components, HL represents the horizontally high-frequent and vertically low-frequent components and HH represents the horizontally high-frequent and the vertically high-frequent components.

After decomposing the image I into 2-D Haar wavelet, we can compute wavelet moment features to capture each candidate text region texture property.

The wavelet energy feature of a pixel (i,j) is defined as:

$$ENG(i,j) = |LH(i,j)| + |HL(i,j)| + |HH(i,j)|. \quad (7)$$

Given a text region candidate R with N_e edge pixels labeled with CL_R, the second and third order wavelet moment can be calculated as

$$M_2(R) = \frac{1}{N_e} \sum_{(i,j)\in R} (ENG(i,j) - MENG(R))^2, \quad (CL(i,j) = CL_R), \quad (8)$$

$$M_2(R) = \frac{1}{N_e^2} \sum_{(i,j)\in R} (ENG(i,j) - MENG(R))^3, \quad (CL(i,j) = CL_R), \quad (9)$$

where MENG(R) is the mean energy feature of region R and

$$MENG(R) = \frac{1}{N_e} \sum_{(i,j)\in R} ENG(i,j), \quad (CL(i,j) = CL_R). \quad (10)$$

Text and non-text regions have different intensity variance and spatial distributions. Wavelet moment features can reflect these differences. Since wavelet energy of contours of text regions are well distributed, large responses of second and third moment features suggest that the region under examination is not text. So we checked the second and third wavelet moment of each candidate text region to verify whether it is a text region or not. The candidate region R is accepted as a true text region if its second moment value falls below T_2 and third moment value falls below T_3, which can be calculated with Eqs. (11) and (12) respectively. The parameters 0.2 and 2.5 used in these two equations are decided by experimental.

$$T_2 = T * 0.2 \quad (11)$$

$$T_3 = T^2 * 2.5 \quad (12)$$

After calculating the second and third order wavelet feature of each region contour, we can filter out false alarms and verify true text regions.

Furthermore, some hole regions appear in many characters, such as “b”, “e”, “P”, “R” etc. Because the hole regions may lead to error result of subsequent text binarization process, we need to remove some hole regions from our text region verification result. Considering spatial relations between the possible internal and external contour of a character, we can easily classify the hole regions circumvented by internal contours. Given one CC G_1, the region circumvented by it will be regarded as a hole only if it meets following two requirements.

- G_1 should be totally surrounded by another CC G_2. Given two CCs G_1 and G_2 with their leftmost, rightmost topmost and bottommost coordinates (G_{l1}, G_{r1}, G_{t1}, G_{b1}), (G_{l2}, G_{r2}, G_{t2}, G_{b2}), if ($G_{l1} \geq G_{l2}$ && $G_{r1} \leq G_{r2}$ && $G_{t1} \geq G_{t2}$ && $G_{b1} \leq G_{b2}$) then G_1 is circumvented by G_2
- The distance between the central points of G_1 and G_2 is small.

Fig. 8 depicts the final text detection result of Fig. 2. As we can see, false alarms and hole regions in Fig. 7 are removed.

2.4 Text Region Binarization

After we verify each region with its texture property, we can binarize each text region separately and feed the result to OCR.

Although human may perceive single character with the same color appearance, the actual pixel colors may vary significantly. So it is necessary to perform color clustering to compensate for these effects.

In each text region, we just want to cluster all colors into two distinctive colors to discriminate between text and other non-text part. So we use a mixture model of Gaussians described with Eq. (13) to depict the color distribution in text regions, where x is an (r, g, b) vector. Because EM algorithm is best suited for fitting Gaussian clusters, we use it to estimate the distribution parameter. The EM algorithm iterates by adjusting the parameters of the Gaussian probability model to maximize the likelihood of data in dataset. The iteration stops when the difference between two successive iterations becomes negligible.

$$P(x) = \sum_{i=1}^{n} \frac{p(i) \times \exp(-\frac{1}{2}(x-\mu_i)^T \sum_i^{-1}(x-\mu_i))}{(2\pi)^{d/2}|\sum_i|^{1/2}} \quad (13)$$

where μ_i is the mean of cluster i, $\sum_i$ is the covariance matrix of cluster i, d is the dimensionality of the data, and d = 3 with color image (rgb component).

In our work, we use the filling algorithm to select two different colors of two pixels inside and outside in text contour to carefully initialize for EM algorithm, but we have

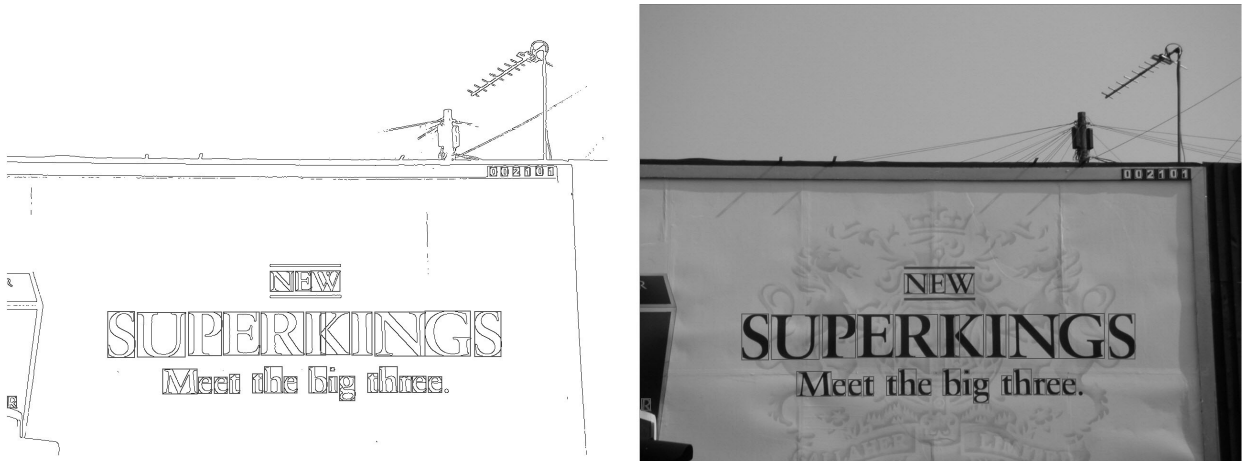


Fig. 8 Text detection result of Fig. 2.

Fig. 9 Text region binarization result.

found that the initialization has little effect on the quality of the resulting clustering process from experiment results.

Upon convergence of the EM algorithm, the two mean vectors can be recorded as the two dominant colors in a text region, i.e. text and non-text part color. Thus we can binarize this text region via measuring the Euclidian distance between each pixel color and two mean color vectors. Figure 9 is the final text region binarization result. So now we can pass the binarized text regions to recognition engine to get better semantic result.

3. Experimental Results

Currently, our algorithm has been implemented in C++ language under Windows-XP on an EPSON Endeavor MT7000 PC equipped with Intel PIV 2.4 GHz processor and 1 G RAM.

As we know, the quantitative evaluation of text detection is still an open research issue due to following two main reasons:

- First, there is no common image database for this task
- Second, no standard measures are defined to report detection result.

In this paper, in order to evaluate the actual performance of our proposed algorithm, we tested 744 real color images which include different types of texts. The image resolution range is between 73*42 and 3072*2048 pixels. Among them, 529 images are from the conference ICDAR 2003 scene text detection competition dataset and other 215 real color images are carefully chosen with a wide variety of background complexity and text types. The ICDAR contest images are further divided into training and testing images. Because our algorithm does not utilize machine learning approach, we included all training and testing images for evaluation. Text appearance varies with different colors, orientation and languages and the character font size (simply the larger value of character width or height) in images ranges from 8 pt to 530 pt.

We divided these 744 test images into two sets. The first set includes 415 images, in which 200 images are from ICDAR contest images. This test set is mainly used to evaluate the robustness of our algorithm, especially the language independency. The second set includes 529 images, all of which are ICDAR contest images, on which we compare the performance of our algorithm and the algorithm proposed in [19].

For measuring accuracy, precision, recall and false alarm rate are used to evaluate our algorithm performance. Recall rate is defined as follows,

$$[\text{Recall Rate}] = \frac{\text{number of detected characters}}{\text{number of characters in image}}$$

False alarm rate is evaluated as follows,

$$[\text{False alarm rate}] = \frac{\text{total pixel count in detected false alarms}}{\text{number of pixels in image}}$$

Precision is calculated as follows,

$$[\text{Precision}] = \frac{\text{number of detected characters}}{\text{total number of detected characters and false alarms}}$$

The classification result of the first test image set is summarized in Table 1. Text detection result of some tested images are shown in Fig. 10, where the left column is the text detection result and the right column is the final binarization result.

Table 2 gives the detection result of our algorithm on

Table 1 Text detection result of the first test image set.

	Character Language			Character Font Size(pixels)			Character Orientation		
	English	Japanese	Chinese	8 200	201 300	301 530	Vertical	Horizontal	Arbitrary
Number of characters	8653	932	542	8270	1131	726	3735	5428	964
	Before Texture Analysis								
Recall rate	91.3%	90.5%	92.7%	91.4%	91.1%	90.5%	90.8%	91.2%	93.8%
False alarm rate	11.5%								
	After Texture Analysis								
Recall rate	91.2%	90.1%	92.4%	91.3%	91.0%	89.7%	90.6%	91.1%	93.5%
False alarm rate	3.2%								

Fig. 10 Examples of text detection and binarization result.

the second test image set. Table 3 lists the detection result of the algorithm proposed in [19] on 367 ICDAR contest images.

Table 1 shows evidence that our algorithm has a high recall rate and low false alarm rate with different text types. Although our texture analysis led to small deficits for recall rate, it greatly reduced the false alarm rate from 11.5% to 3.2%. Some false alarms left because of their strong texture features, high contrast and resemble text-like attributes. So we plan to combine recognition engine into detection process to further reduce the false alarms in future. Experimental results also demonstrate that our algorithm can detect isolated letters, such as an isolated character "R" detected in Fig. 9 and an isolated character "3" detected in Fig. 10, which have no neighboring text region.

In addition, because some Chinese characters consist of more than two CCs, our algorithm may segment an individual Chinese character into several regions and combining recognition engine into detection process is also helpful to merge some fragmented components into a single character.

Fig. 11 Examples of text detection errors.

Table 2 Text detection result of the first test image set.

	Precision	Recall rate
529 ICDAR contest images	73.4%	79.3%

Table 3 Text detection result of the first test image set.

		Precision	Recall rate
367 ICDAR contest images	Training	64%	66.9%
	Testing	59.2%	73.2%

With the second test image set, our algorithm achieved recall rate of 79.3% and precision of 73.4%. Table 3 shows the evidence that our algorithm has much higher average recall rate and precision than the algorithm proposed in [19].

The performance of our algorithm for the second test image set is much lowered than that of the first test image set because some images in the second set have very much complex background with significantly perturbed intensities and texts in these images have very low contrast and have much complicated connection with other non-text objects. Figure 11 shows some incorrectly text detection result. The errors occurred because some texts are strongly connected with other objects in the image and some non-text objects have strong text like texture patterns.

Experimental results also show that the binarization procedure with EM algorithm is also quite usable. Even when the color distribution inside a region is unimodal, such as characters like "l", "_" etc., the two mean vectors become nearly coincident, which makes the binarization process more robust.

The speed of our text detection algorithm is a function of the size and complexity of the color images. The speed of binarization process is determined by the size and number of characters in color images. Our algorithm usually can detect texts in one 24-bit color image of size 512*512 pixels within 0.5 second on an EPSON Endeavor MT7000 PC. With the image shown in Fig. 2, our text detection and binarization process take 0.762 seconds and 3.752 seconds respectively.

In images, there may exist some connected characters, which are components of connected multiple characters. After text region binarization, touching characters need further process to separate each single character. So we intend to integrate our former work [20] into our algorithm to segment touching characters.

4. Conclusions

In this paper, we propose a hybrid approach, which combines CC based and texture based methods, to detect variant texts in color images. First, an elaborate image edge extraction algorithm and CC analysis are used to extract candidate text regions. Then texture feature in wavelet domain is explored to identify text regions from candidate regions. After text region verification, EM algorithm is introduced to binarize all text regions to prepare data for OCR. Because gradient, geometrical and texture features of text contours are all invariant to character size, orientation, language and color, our algorithm can detect texts in different appearance.

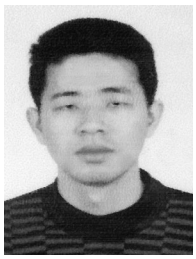
Acknowledgments

This work was supported by fund from the MEXT via Kitakyushu innovative cluster project.

References

[1] C. Li, X. Ding, and Y. Wu, "Automatic text location in natural scene images," Proc. Sixth International Conference on Document Analysis and Recognition, pp.1069–1073, Sept. 2001.
[2] J. Zhou and D. Lopresti, "Extracting text from WWW images," Proc. Fourth International Conference on Document Analysis and Recognition, vol.1, pp.248–252, Aug. 1997.
[3] D. Chen, H. Bourlard, and J.P. Thiran, "Text identification in complex background using SVM," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2, pp.621–626, Dec. 2001.

[4] Y. Zhong, K. Karu, and A.K. Jain, "Locating text in complex color images," Pattern Recognit., vol.28, no.10, pp.1523–1536, Oct. 1995.
[5] A.K. Jain and B. Yu, "Automatic text location in images and video frames," Proc. Fourteenth International Conference on Pattern Recognition, vol.2, pp.1497–1499, 16-20, Aug. 1998.
[6] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," IEEE Trans. Circuits Syst. Video Technol., vol.12, no.4, pp.256–268, April 2002.
[7] Y. Zhong, H. Zhang, and A.K. Jain, "Automatic caption localization in compressed video," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.4, pp.385–392, April 2000.
[8] X. Tang, X. Gao, J. Liu, and H. Zhang, "A spatial-temporal approach for video caption detection and recognition," IEEE Trans. Neural Netw., vol.13, no.4, pp.961–971, July 2002.
[9] K. In Kim, K. Jung, and J. Hyung, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.12, pp.1631–1639, Dec. 2003.
[10] M. Fujii and W.J.R. Hoefer, "Filed-singularity correction in 2-D time-domain Haar-wavelet modeling of waveguide components," IEEE Trans. Microw. Theory Tech., vol.49, no.4, pp.685–691, April 2001.
[11] Q. Ye, W. Gao, W. Wang, and W. Zeng, "A robust text detection algorithm in images and video frames," Proc. 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific-Rim Conference On Multimedia, vol.2, pp.802–806, Dec. 2003.
[12] K. Sobottka, H. Bunke, and H. Kronenberg, "Identification of text on colored book and journal covers," Proc. Fifth International Conference on Document Analysis and Recognition, pp.57–62, Sept. 1999.
[13] Y. Zheng, H. Li, and D. Doermann, "Machine printed text and handwriting identification in noisy document images," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.3, pp.337–353, March 2004.
[14] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," Proc. IEEE International Conference on Computer Vision, vol.1, pp.454–461, July 2001.
[15] O. Hori and T. Mita, "A robust video text extraction method for character recognition," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J84-D-II, no.8, pp.1800–1808, Aug. 2001.
[16] H.-C. Lee and D.R. Cok, "Detecting boundaries in a vector field," IEEE Trans. Signal Process., vol.39, no.5, pp.1181–1194, May 1991.
[17] N. Jin and Y.Y. Tang, "Text area localization under complex-background using wavelet decomposition," Proc. Sixth International Conference on Document Analysis and Recognition, pp.1126–1130, Sept. 2001.
[18] A. Mojsilovic, M.V. Popovic, and D.M. Rackov, "On the selection of an optimal wavelet basis for texture characterization," IEEE Trans. Image Process., vol.9, no.12, pp.2043–2050, Dec. 2000.
[19] K.C. Kim, H.R. Byun, Y.J. Song, Y.W. Choi, S.Y. Chi, K.K. Kim, and Y.K. Chung, "Scene text extraction in natural scene images using hierarchical feature combining and verification," Proc. 17th International Conference on Pattern Recognition, vol.2, pp.679–682, 2004.
[20] Y. Liu, Y. Luo, F. Liu, and Z. Qiu, "A novel approach of segmenting touching and kerned characters," Proc. 8th International Conference on Neural Information Processing, vol.3, pp.1603–1606, Oct. 2001.



Yangxing Liu is a Ph.D. student of the Graduate School of Information, Production and Systems at Waseda University, Japan. He received the M.E. degree from the Department of Automation, Tsinghua University, China, in 2002. His research interests include pattern recognition, computer vision and VLSI design.

Satoshi Goto was born on January 3rd, 1945 in Hiroshima, Japan. He received the B.E. degree and the M.E. degree in Electronics and Communication Engineering from Waseda University in 1968 and 1970, respectively. He also received the Dr. of Engineering from the same university in 1981. He is IEEE fellow, Member of Academy Engineering Society of Japan, Professor of Waseda University. His research interests include LSI system and multimedia system.

Takeshi Ikenaga received the B.E. and M.E. degrees in electrical engineering and the Ph.D degree in information computer science from Waseda University, Japan, in 1988, 1990, and 2002, respectively. He joined LSI Laboratories, NTT Corporation in 1990, where he has been undertaking research on the design and test methodologies for high-performance ASICs, a real-time MPEG2 encoder chipset, and a highly parallel LSI system design for image processing. He is presently an associate professor in the system LSI field of the Graduate School of Information, Production and Systems, Waseda University. His current interests are application SoCs for image, security and network processing. Dr. Ikenaga is a member of the IPSJ and the IEEE. He received the IEICE Research Encouragement Award in 1992.