Pattern Recognition 46 (2013) 131–140

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A novel ring radius transform for video character reconstruction

Palaiahnakote Shivakumara [a,*], Trung Quy Phan [a], Souvik Bhowmick [a], Chew Lim Tan [a], Umapada Pal [b]

[a] School of Computing, National University of Singapore, Singapore
[b] Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, India

ARTICLE INFO

Article history:
Received 10 November 2011
Received in revised form
22 June 2012
Accepted 12 July 2012
Available online 1 August 2012

Keywords:
Ring radius transform
Video document processing
Gap filling
Video character reconstruction

ABSTRACT

Character recognition in video is a challenging task because low resolution and complex background of video cause disconnections, loss of information, loss of shapes of the characters etc. In this paper, we introduce a novel ring radius transform (RRT) and the concept of medial pixels on characters with broken contours in the edge domain for reconstruction. For each pixel, the RRT assigns a value which is the distance to the nearest edge pixel. The medial pixels are those which have the maximum radius values in their neighborhood. We demonstrate the application of these concepts in the problem of character reconstruction to improve the character recognition rate in video images. With ring radius transform and medial pixels, our approach exploits the symmetry information between the inner and outer contours of a broken character to reconstruct the gaps. Experimental results and comparison with two existing methods show that the proposed method outperforms the existing methods in terms of measures such as relative error and character recognition rate.

1. Introduction

The recognition of characters has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence [1]. In terms of application, with the proliferation of videos on the Internet, there is an increasing demand for video retrieval. In the early '1960s', optical character recognition was deemed as one of the first successful applications of pattern recognition, and today, for simple tasks with clean and well formed data, character recognition in document analysis is viewed as a solved problem [2]. However, automatic recognition of text from natural scene images is still an active field in the document analysis due to the variety of texts, different font sizes, orientations and occlusion [3–6]. To achieve better character recognition rate for the text in natural scene images, set of methods have been proposed based on Gabor features and linear discriminate analysis [3], cross ratio spectrum and dynamic time wrapping [4], conditional random field [5], Markov random field [6] in the literature. According to the literature on natural scene character recognition, these methods have so far achieved only recognition rates between 60% and 72% [4]. This poor accuracy is due to the complex background in natural scene images. Experiments show that applying conventional character recognition methods directly on video text frames leads to poor recognition rate, typically from 0% to 45% [1,7]. This is because of several unfavorable characteristics of video such as high variability in fonts, font sizes and orientations, broken characters due to occlusion, perspective distortion, color bleeding, disconnections due to low resolution and complex background [8]. In addition to this, concavity, holes and complicated shapes make the problem more complex. For example, Fig. 1 shows the complexities of a video character compared to a document character. As a result, OCR does not work well for video images.

As a result of the above problems, methods based on connected component (CC) analysis are not good enough to handle the problems of video characters. Therefore, in the present paper, we focus on reconstruction of character contours to increase the recognition rate because contours are important features which preserve the shapes of the characters and help in reconstruction. The work is motivated by the following considerations: (i) we only have to deal with a limited set of well-defined shapes for the present study, (ii) characters' shapes are sufficiently challenging and (iii) optical character recognition (OCR) allows us to have an objective means of assessment for the reconstruction results. Hence, the scope of the work is limited to character contour reconstruction from the broken contours. One way to reconstruct the broken contour is to exploit the symmetry of the character image shape, e.g. between the left- and right-hand sides of the contour or between the inner and outer contours of the same character image. Contour reconstruction based on symmetry of contours is motivated by the way the human visual system works. Research has shown that when an object is occluded, human

* Corresponding author. Tel.: +65 83050895.
E-mail addresses: shiva@comp.nus.edu.sg, hudempsk@yahoo.com (P. Shivakumara), phanquyt@comp.nus.edu.sg (T.Q. Phan), sou.bhowmick@gmail.com (S. Bhowmick), tancl@comp.nus.edu.sg (C.L. Tan), umapada@isical.ac.in (U. Pal).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.07.008

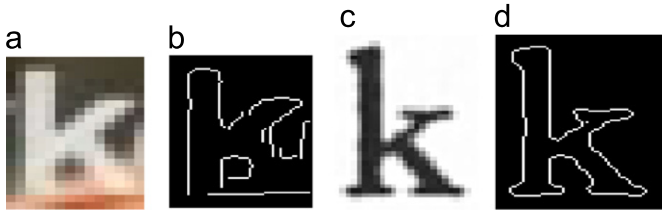


Fig. 1. Characters in videos (a) have poorer resolution, lower contrast and more complex background than characters in document images (c). As a result, the former often has broken contours (b) while the later has complete contours (d). Contours of (a) and (c) are shown in (b) and (d), respectively.

beings are still able to recognize it by interpolating the observed incomplete contour with their knowledge about the shapes of the objects that they have seen before [9]. This symmetry can be observed in many kinds of objects, from real world objects to industrial objects and even organs in the human body.

2. Previous work

Video character recognition methods use either enhancement by integrating temporal frames or by improving binarization or by filling gaps in the character contours to address the issue of video character recognition. In this work, we choose gaps filling to improve recognition rather than enhancement because for enhancement there is no validation method and it is hard to justify that the proposed enhancement criteria work for all kinds of images.

The method [7] proposes a multi-hypothesis approach to handle the problems caused by the complex background and unknown gray scale distribution to recognize the video characters. The temporal information has been exploited for video character recognition in the method [1] which involves Monte Carlo video text segmentation step and recognition results voting step. To recognize Chinese and Korean characters, the method [8] proposes a holistic approach which extracts global information of character as well as component wise local information of the characters. Several methods [10,11] are proposed for caption text recognition in video using measures of accumulated gradient and morphological processing and Fuzzy-clustering neural network classifier for which the methods extract spatial-temporal information of the video. The above recognition methods are good for caption text, artificial text and text of big fonts with good quality since the extracted features infer the shape of the characters in CC analysis. However, these features will not work for scene characters because binarization methods may not preserve the shape of the characters and lead to incomplete shape of the characters due to disconnections and loss of information.

The methods for binarization on low-quality text using Markov Random Field Model are proposed in [12,13] where it is shown that the methods preserve the shape of the character without losing significant information for the low quality character. However, the methods require training samples to train the classifier and are sensitive to complex background of video images. An improved method for restoring text information through binarization is also proposed in [14] where the method focuses on scanned handwritten document images but not video images. Recently, a double threshold image binarization method based on edge detector is developed in [15] where it is shown that the method works for low contrast, non-uniform illumination and complex background. However, due to double thresholding the method loses significant information and hence the methods are not good enough to tackle the video images.

There are other works which propose specific binarization methods to deal with video characters in order to improve their recognition rate. The method [16] uses corner points to identify candidate text regions and color values to separate text and non-text information. For the same purpose, the method [17] proposes convolutional neural network classifier with training samples to perform binarization. These methods are good if the character pixels have high contrast without disconnections and loss of information. The method [18] aims to address the disconnection problem by proposing a modified flood fill algorithm for edge maps of video character images to fill small gaps in the character images. All these methods cannot totally prevent the problem of broken characters due to the low resolution, low contrast and complex background nature of video images.

In the document analysis community, there are methods that fill small gaps caused by degradations and distortions in contour to improve character recognition. This is often done by utilizing the probability of a text pixel based on its neighbors and filling in the gap if required. Wang and Yan [19] proposed a method for mending broken handwritten characters. Skeletons are used to analyze the structures of the broken segments. Each skeleton end point of a CC is then extended along its continual direction to connect to another end point of another CC. In a similar approach for broken handwritten digits, Yu and Yan [20] identify structural points, e.g. convex and valley points, of each CC. For each pair of neighboring CCs, the pair of structural points that have the minimum distance is considered for reconstruction. Different from the previous two methods, Allier and Emptoz [21] and Allier et al. [22] use active contour to reconstruct broken characters in degraded documents. Given a binary image, features extracted from Gabor filters are used for template matching. The lack of external forces at gap regions is compensated by adding gradient vector flow forces extracted from the corresponding region of the template. This compensation made the snake converge to the character contour instead of going inside it. As the above methods are designed for document images, they rely heavily on CC analysis. However, this is not suitable for video images because it is extremely difficult to extract characters as complete CCs.

Based on the above considerations, we introduce a novel concept of ring radius transform (RRT) and medial pixels to fill in the gaps of a broken character based on the symmetry between its inner and outer contours. For example, if a gap occurs on one contour while the other contour is fully preserved, it can be filled in by “copying” the pixels from the corresponding region of the other contour. As another example, if a gap occurs on both contours, it may still possible to recover this gap by using the information from the neighboring regions of the gap. “Copying”, as mentioned above, is achieved by introducing the concepts of RRT and medial pixels. For each pixel, RRT assigns a value which is the distance to the nearest edge pixel. The medial pixels are defined as the pixels at the middle of the inner and outer contours. In terms of radius values, they have the maximum values in their neighborhood because they are close to neither of the contours.

There are two main contributions in this paper. First, we propose to use the symmetry information between the inner and outer contours to recover the gaps. This is a departure from the traditional approach of considering a broken character as a set of CCs and trying to connect these components. The second contribution lies in the concepts of RRT and medial pixels. The stroke width transform and medial axis concepts are explored for natural scene text detection in [23]. This method computes stroke width based on gradient information and it is for high-resolution camera images while our idea is based on distance transform and for video character reconstruction. Although there are related ideas in the literature, this is the first attempt to use such concepts for reconstruction of broken character image contours. The key difference between our method and other reconstruction methods in the literature is that our reconstruction method is done directly on the character contours instead of the character pixels (in the form of CCs).

3. Proposed approach

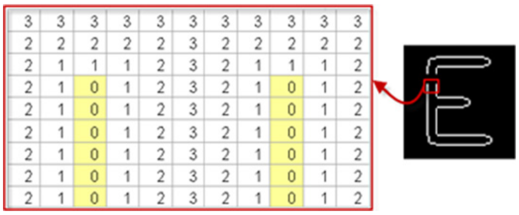
For reconstruction of character contour, we use our previous method in [24] for character segmentation from the video text line extracted by the text detection method [25]. This method treats character segmentation as a least cost path finding problem and it allows curved segmentation paths. Therefore, the method segments character properly even if there is a touching and overlapping characters due to low contrast and complex background. Gradient vector flow is used to identify the candidate cut pixels. Then two-pass path finding algorithm is proposed for identifying true cuts and removing false cuts. In addition, the method has ability to segments the characters from the multi-oriented text line. Therefore, in this work, we treat non-horizontal character as same as horizontal character. Thus we use the output of segmentation method as input for character reconstruction. This section is divided into two subsections. In the first subsection, we introduce a novel idea of RRT and medial axis for reconstruction of individual character images obtained by the character segmentation method and in the second subsection, we present detailed filling algorithmic steps for character reconstruction.

3.1. Ring radius transform and medial pixels

The input for RRT is the edge map of the segmented character image. For a given edge map, RRT produces a radius map of the same size, in which each pixel is assigned a value according to the distance to the nearest edge pixel. The radius value is defined mathematically as follows:

$$rad(p) = \min_{q:f(q)=1} dist(p,q) \tag{1}$$

Here $rad(x)$ returns the radius value of a pixel x in f, a binary image where edge pixels and background pixels are assigned values 1 and 0, respectively. $dist(x, y)$ is a distance function between two pixels x, y. Fig. 2 shows a sample radius map for the gap in the character on the right side. One can notice from Fig. 2 that the values marked by yellow color are text pixel having radius zero of the ring. It is also observed the values between two zero radii marked by yellow color that increase from zero (left text pixel) to highest radius value (3) (we call the highest radius as medial axis value) and again decrease in the same way to reach the zero radius value (right text pixel). Among the values returned in the radius map, we are interested in the medial pixels, i.e. those are at the middle of the inner and outer contours and thus have the maximum radius values in their neighborhood. *Horizontal medial pixels* (*HMP*) are the peak pixels compared to the neighboring pixels on the same row while *vertical medial pixels* (*VMP*) are defined with respect to the neighboring pixels on the same column (Fig. 2). Medial pixels are useful for analyzing character image regions with almost constant thickness. The medial pixels would lie along the center axes of the contours and have similar radius values, which are roughly half of the region widths. Potential gaps can then be identified by checking for contour pixels on both sides of the medial pixels based on the radius values (see Section 3.3.2 for more details).



3	3	3	3	3	3	3	3	3	3	3
2	2	2	2	2	3	2	2	2	2	2
2	1	1	1	2	3	2	1	1	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2

Fig. 2. Sample values in the radius map (using the chessboard distance function). The highlighted values (yellow color) are the text (white) pixels within the window. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

It is clear from the above formulation that no character-specific features have been used. In other words, these concepts generalize well to any objects whose contours possess the symmetry property. Besides, none of the related transforms in the literature has been used in this way.

3.2. Character reconstruction method

The proposed method consists of six steps. In the first step, we extract the character contours from the input grayscale images. Based on these contours, the medial pixels are identified in the second step. The third step then uses the symmetry information from the medial pixels to reconstruct horizontal and vertical gaps. The fourth uses both vertical and horizontal medial axes iteratively to fill large gaps. The fifth step fills the gaps in the outer contour. The sixth steps fill in all the remaining small gaps. These steps can be seen in Fig. 3 where the flow diagram of the character reconstruction is given.

3.2.1. Extracting character contours

The purpose of this step is to extract character contours from the input grayscale image. In order to be readable, a character should have reasonable contrast with the background. Therefore, we propose to use the *Normalized Absolute Gradient* (NAG) based on horizontal gradient information obtained by the convolution with the $[-1, 1]$ mask

$$gx\_abs(x,y) = |h(x+1,y) - h(x,y)| \tag{2}$$

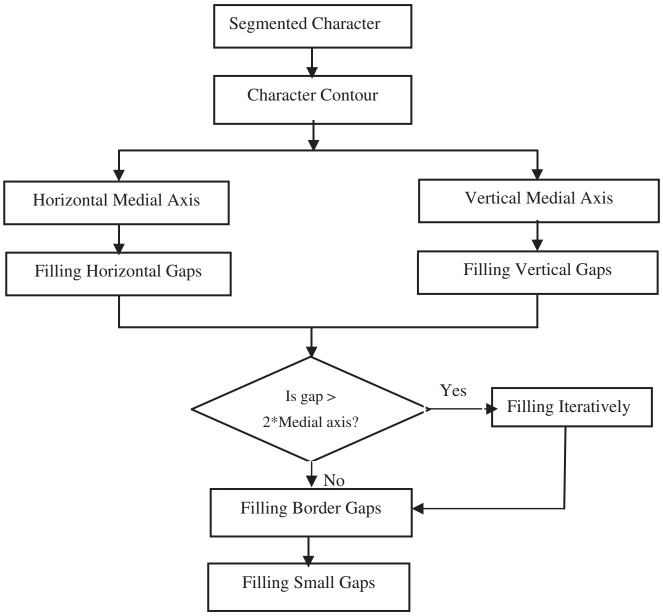


Fig. 3. Flow diagram of character reconstruction method for the segmented character.

$$gx\_norm(x,y) = \frac{gx\_abs(x,y) - \min(gx\_abs)}{\max(gx_{abs}) - \min(gx\_abs)} \qquad (3)$$

Here h is the input image, $min(gx\_abs)$ and $max(gx\_abs)$ return the minimum and maximum values of gx_abs. $\forall x,\ y.\ gx\_norm(x,\ y) \in [0,1]$.

It is expected that pixels on character contours will have higher gradient values than those in the background as it can be seen in Fig. 4(b) for the image shown in Fig. 4(a). Fig. 4(b) shows that NAG values of the contour pixels are brighter than other pixels. We use Fuzzy c-means clustering to classify all pixels into two clusters: text and non-text. The advantage of Fuzzy c-means is that it allows a data point to belong to more than one cluster. It is shown in [26] that Fuzzy c-means gives better results for text pixel classification.

After Fuzzy c-means clustering, we have two centroids for text and non-text feature values. Depending on those two centroids we binarize the character image to get the text cluster image. Although the text cluster contains most of the contour pixels, it may still miss some contour pixels of lower contrast as shown in Fig. 4(c). To recover them, we take the union of the text cluster in Fig. 4(c) and the Canny edge map shown in Fig. 4(d). The canny edge map is obtained by performing Canny edge operator on the input image shown in Fig. 4(a). It is also true that sometimes Canny edge map loses text information due to undesirable characteristics of video. One such example is shown in Fig. 4(d) where an edge on the left side of the character “N” is missing. This shows that neither text cluster alone nor Canny edge map alone is sufficient to extract the complete contour of the character. Therefore, in this work, we propose an union operation which results in fewer gaps in the contour. The union operation output is shown in Fig. 4(e). Finally, thinning is performed to reduce the contour thickness to one pixel as shown in Fig. 4(f).

Fig. 4 shows the intermediate results of various steps in this section. The missing gaps of both the inner and outer contours are reconstructed in the next sections.

3.2.2. *Identifying horizontal and vertical medial pixels*

In this step, we apply RRT to the initial character contour image given by the previous step using the chessboard distance function

$$dist(p,p') = \max(|p.x - p'.x|, |p.y - p'.y|) \qquad (4)$$

In other words, squares centered at p are used instead of rings, for ease of implementation. The output of Eq. (4) is shown in Fig. 5 where the horizontal and vertical medial axis values are indicated on the left and right sides, respectively. Medial pixels are then identified as described in Section 3.1. The final medial axis with respect to the horizontal and vertical medial axis pixels can be seen in Fig. 6 on the left and right sides, respectively.

Medial pixels provide useful information about the symmetry between the inner and outer contours of a character. For example, suppose that a gap occurs at the outer contour while the inner contour is fully preserved during the extraction step. The medial pixels near the gap will have similar values, which are their distances to the inner contour. By traversing those distances in the opposite direction (towards the outer contour), we will be able to detect that some contour pixels are missing.

As another example, suppose that both the inner and outer contours are broken at a particular region of a vertical stroke. If the nearest medial pixels above and below the gap have the same value, the regions immediately above and below the gap are likely to belong to the same stroke because of the same stroke width. The gap can then be reconstructed based on these two regions. Therefore, medial pixels help us to utilize not only the symmetry information but also the similarities between nearby regions. These information is used to fill in the horizontal and vertical gaps in the next section.

3.2.3. *Filling horizontal and vertical gaps*

The horizontal gaps will be filled using vertical medial pixels and vice versa. Since the two cases are symmetric, we will only discuss the first one in detail in this section.

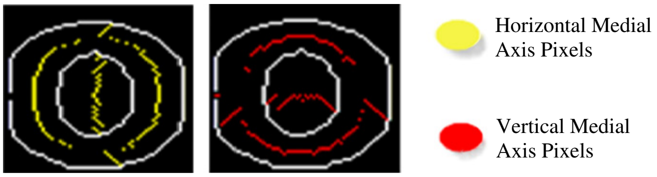


Fig. 6. Horizontal medial axis (left) and vertical medial axis (right).

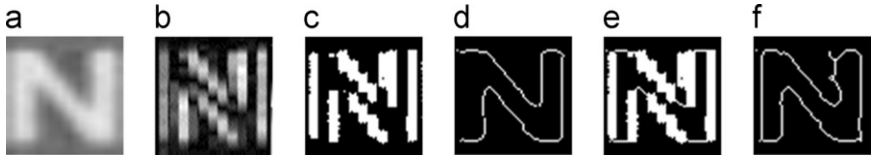
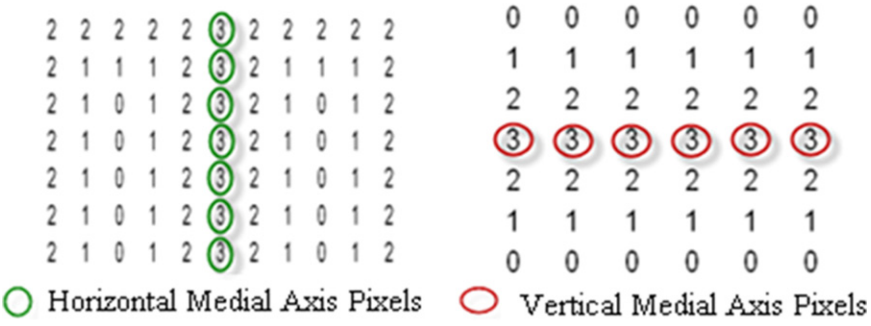


Fig. 4. Various steps of extracting character contours from grayscale input images. (a) Input, (b) NAG of (a), (c) text cluster of (b), (d) Canny of (a), (e) union of (c) and (d), and (f) thinning of (e).



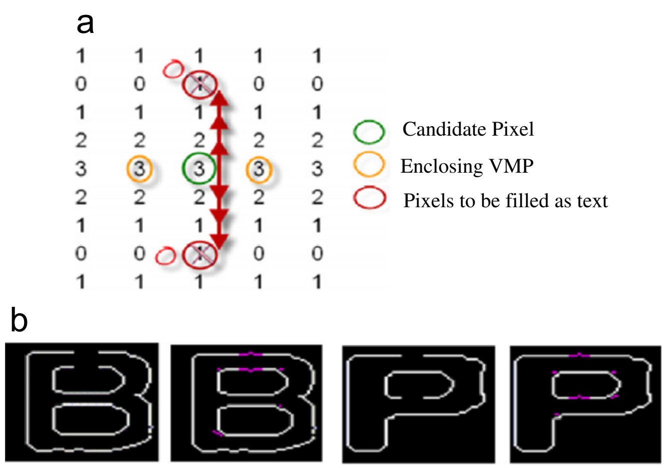
2	2	2	2	2	3	2	2	2	2	2
2	1	1	1	2	3	2	1	1	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2
2	1	0	1	2	3	2	1	0	1	2

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
2	2	2	2	2	2
1	1	1	1	1	1
0	0	0	0	0	0

○ Horizontal Medial Axis Pixels ○ Vertical Medial Axis Pixels

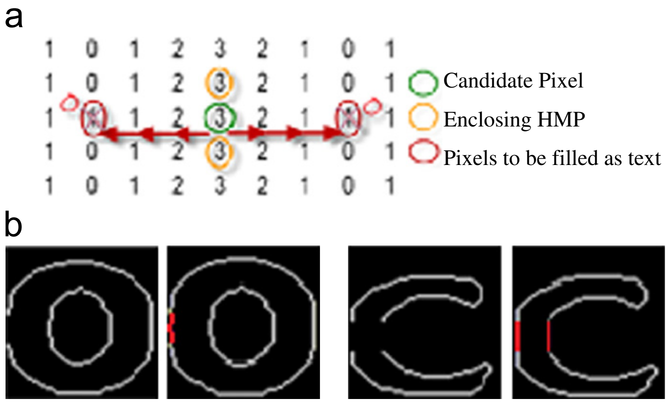
Fig. 5. Horizontal and vertical medial axis pixels marked by green and red color. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

For every pixel p (Candidate pixel) in the radius map generated in the previous step as shown in Fig. 7(a), we will find two nearest vertical medial axis pixels (VMP), one on the left of the pixel and other the right of the pixel as shown by orange color in Fig. 7(a). If the original pixel and the two VMPs have exactly the same value r, it indicates that we are likely to be in the middle of a horizontal stroke of a character due to the constant thickness. We will check two pixels, $(p.x-r, p.y)$ and $(p.x+r, p.y)$, and mark them as text pixels if they are currently classified as non-text as shown in pixel to be filled as text by red color in Fig. 7(a). The horizontal gap is thus filled as shown in the two examples in Fig. 7(b). In the same way, we use horizontal medial axis pixels (HMP) to fill the vertical gap as shown illustration in Fig. 8(a) and the sample vertical filled results are shown in Fig. 8(b).



1	1	1	1	1
0	0	~~1~~	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
2	2	2	2	2
1	1	1	1	1
0	0	~~1~~	0	0
1	1	1	1	1

Fig. 7. Sample results of horizontal gap filling. (a) Illustration of horizontal gap filling based on VMP and (b) horizontal gaps filled. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)



1	0	1	2	3	2	1	0	1
1	0	1	2	3	2	1	0	1
1	~~1~~	1	2	3	2	1	~~1~~	1
1	0	1	2	3	2	1	0	1
1	0	1	2	3	2	1	0	1

Fig. 8. Sample results of vertical gap filling. (a) Illustration of vertical gap filling based on HMP and (b) vertical gaps.

3.2.4. Filling large gaps iteratively

The above step fills horizontally and vertically if the contour of a character has gaps of size $2\times r$ or less, where r is the medial axis pixel value. This is the advantage of the RRT method in filling gap compared to the other gap filling methods such as smoothing and morphological processing in document analysis. If a gap exceeds the size mentioned above then it is considered as a large gap and this gap is filled iteratively by horizontal and the vertical gap filling algorithms. For large gap, the RRT is computed at each iteration to obtain the medial axis in the gap. It is observed that if there is a large gap as shown in Fig. 9(a), the medial axis can be formed even in the gap. In Fig. 9(b) we observe that the medial axis (marked in yellow) is extended a few pixels away down from the upper end pixels and a few pixels above the lower end pixels. In this situation, the horizontal and the vertical gap filling algorithms utilize the medial axis information to close in the gap. In Fig. 9(c) we see that the large gap has become a small gap (less than $2\times r$). As it is explained in Section 3.2.3, this small gap gives medial axis information as shown in Fig. 9(d) which allows the horizontal filling algorithm to fill up the gap automatically. In this way, the horizontal and the vertical filling algorithms fill a large gap using extended medial axis information iteratively until the algorithm finds no gap (connected component) as shown in Fig. 9(e) where the gap is filled in the second iteration.

3.2.5. Filling border gaps

It is true that the above horizontal and vertical filling algorithms fill only gaps in horizontal and vertical direction but not in diagonal direction which exists at the corners of the contours. Therefore, in this section, we propose a criterion to fill any gaps on the outer contour of the character including gaps at corners, i.e. the border of the character. In this step, we describe the process of filling border gaps of the contours based on the radius information in both the horizontal and vertical directions.

Every non-text pixel which is near to the boundary of a character is represented by a high negative value in the radius map of the contour because for these pixels the boundary is nearer than the edge pixel. As a result, for non-text pixels, the negative values are assigned in the radius map of the contour. In other words, background of non-character area is represented by high negative values as shown in Fig. 10(a) (values marked in green color). From the medial axis values, the algorithm finds the outer contour and checks for any high negative values in the outer contour. It then fills the gap based on the criterion used in horizontal and vertical filling algorithms. The sample results are shown in Fig. 10(b) where one can notice that the algorithm fills gaps at corner and other small gaps that are missed by the horizontal and vertical filling algorithms. Fig. 10(b) shows that the gaps on the inner contour have not been filled by the algorithm as this algorithm fills gaps on the outer contour but not gaps on the inner contour. This is because once the algorithm fills the gaps on the outer contour, filling in the gaps on the inner contour becomes easy for the proposed method. Note that for the character ‘R’ shown in Fig. 10(b), the algorithm also fills non-gap

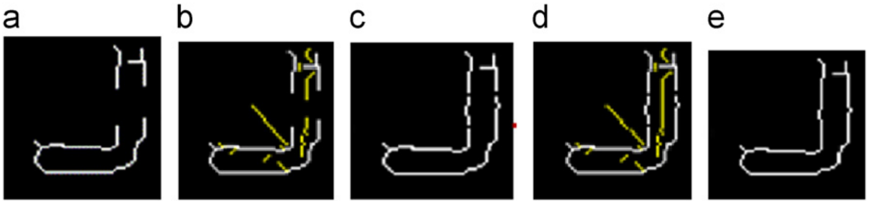


Fig. 9. Iterative filling helps to mend large gaps. (a) Input, (b) medial axis, (c) first iteration results, (d) medial axis, and (e) second iteration results. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

on the right side of the character. This causes problems for the next step of the algorithm. Hence, we perform preprocessing to remove such extra noisy information.

3.2.6. *Filling small gaps*

Most of the big gaps have been filled in the previous steps. The purpose of this step is to handle the remaining gaps, most of which are quite small and are missed by the above step of the algorithms.

We have found that the result of the previous steps may contain extra information, e.g. small branches, loops and blobs, which should be removed before filling in the gaps. Small branches are removed as follows: for each CC (usually a contour or part of a contour), only the longest possible 8-connected path is retained; all other pixels are discarded as shown in Fig. 11(a). Loops and blobs are removed if their lengths and areas are below certain thresholds, which are determined adaptively based on the estimated stroke widths (twice the radius values) as shown in Fig. 11(b) and (c).

It is observed that if a character is not broken, there will only be closed contours (e.g. 1 contour for ‘Y’, 2 contours for ‘A’ and 3 contours for ‘B’) and no end points at all. Therefore, after the image has been cleaned up, the small gaps are identified by looking for end points. A gap often creates two end points, which motivates us to propose the *mutual nearest neighbors* concept to not only find the end points but also pair them together. *p1* and *p2* are mutual nearest neighbors if *p1* is the nearest edge pixel of *p2* and vice versa. Each pair of end points is connected by a straight line to fill in the gap between the two points as shown in Fig. 12. The preprocessing steps such as removing small branches ensure that the end points are true end points and thus we avoid filling in false gaps. Fig. 12 shows the small gap filling algorithm fills even large gaps also as shown one example for the character ‘d’. It is observed from Fig. 12 that the small gap filling algorithm does not preserve the shape of the contour while filling gaps. This is the main drawback of this algorithm. However, it does not affect the final recognition rate much because this algorithm is used in this work only for filling small gaps that remain after horizontal and vertical filling algorithms but not for large gaps.

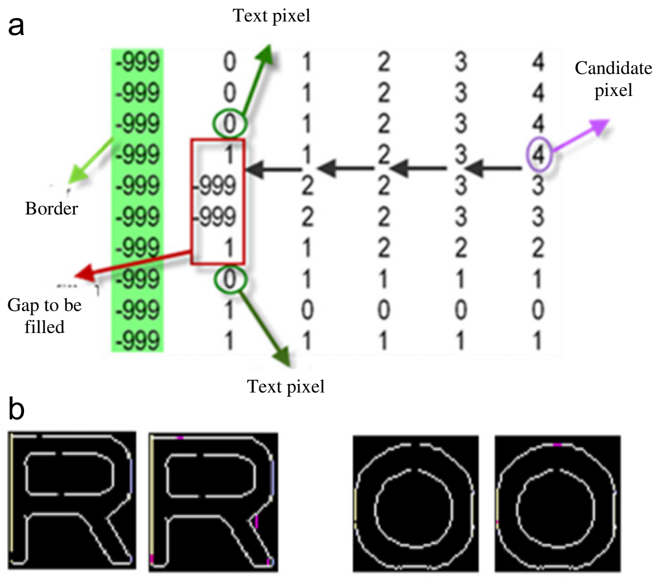


Fig. 10. Sample results of border gap filling. (a) Illustration of border gap filling and (b) border gaps filled. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Finally, before the reconstructed character is sent for recognition, flood filling and inversion are done to get a solid black character on a white background as shown in third column in Fig. 13.

4. Experimental results

Since there is no standard dataset for character reconstruction, we have selected 131 character images for our own dataset. The characters are extracted from TRECVID videos of different categories, e.g. news, movies and commercials. In this work, we focus on English characters and numbers. We check character contour extraction results on our video character dataset and then we select a variety of broken characters as input and neglect characters which have no gaps. In addition to this, we add a few manually cut characters to create large gaps on the contours to test the effectiveness of the method. The performance of the proposed method is measured at two levels that are relative error for measuring reconstruction results as qualitative measurement and the character recognition rate (CRR) before and after reconstruction using Tesseract [27], Google’s OCR engine.

For comparison purpose, we have implemented two existing methods, one from the enhancement approach and the other from the reconstruction approach which fill the gaps in the contour of the characters. Method [18], denoted as *flood fill-based method*, performs binarization on the input character by modifying the traditional flood fill algorithm. Method [22], denoted as *active contour-based method*, reconstructs the character contour from ideal images using gradient vector flow snake.

The original active contour-based method requires a set of ideal images, or templates, for snake initialization. However, as we are using the method for reconstruction prior to recognition,

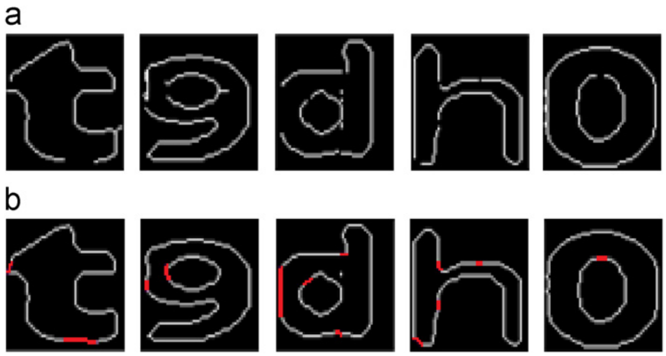


Fig. 12. Sample results of small gap filling. (a) Input images and (b) gaps filled.

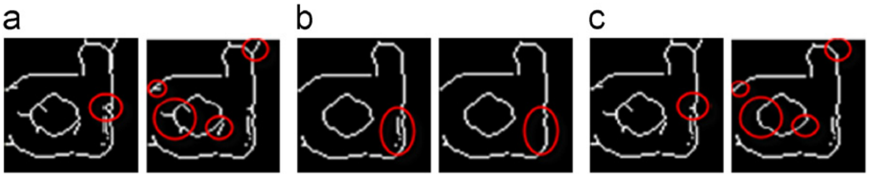


Fig. 11. Preprocessing steps during small gap filling. (a) Removal of small loops, (b) removal of small branches, and (c) removal of small blobs.

the ideal template for the character to be recognized is not known. Thus, we have modified the active contour-based method to use only information from the input broken character. That is, the template is now the input image itself, except that the end points have not been connected using the mutual nearest neighbors concept of Section 3.2.6. The reason for this simple preprocessing step is to get closed contours, which are required for snake initialization.

4.1. Sample reconstruction results on character images

Fig. 13 shows few sample results of the existing methods (flood fill-based and active contour-based) and the proposed method. The flood fill-based method works well for a few images (1, 3, 7 and 8) where the shape is preserved and small gaps exist on contours. The main reason for poor reconstruction in the remaining images is that the modified stopping criteria used in this method are still not

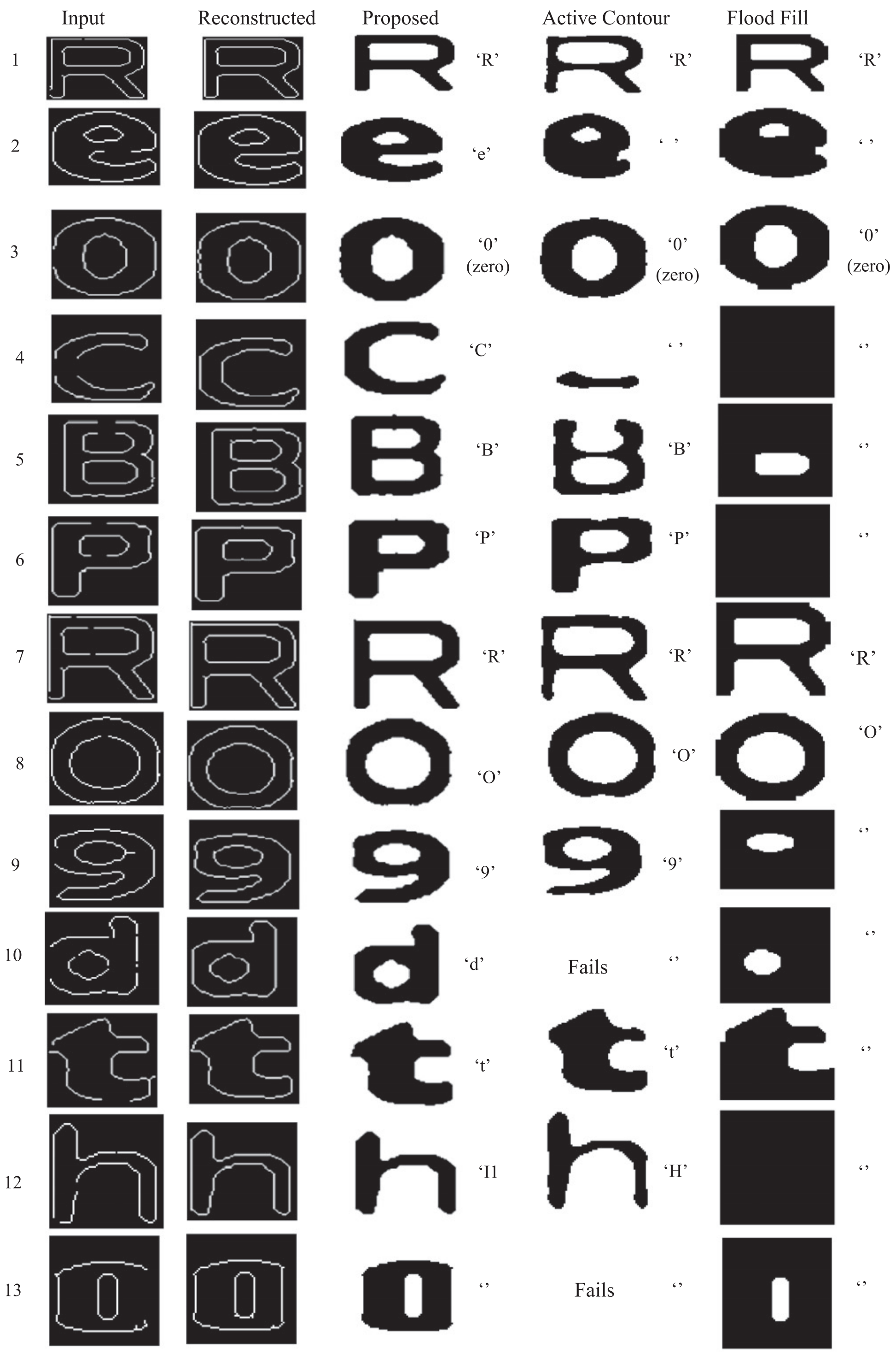


Fig. 13. Recognition results of the proposed and existing methods. Recognition results are shown within quote in the right side of the respective character. Here the symbol ‘ ‘ indicates that the OCR engines returns an empty string for the input character where OCR engine fails to recognizes and “Fails” shows the methods gives nothing.

sufficient to prevent the filling of parts of the background. The active contour-based method reconstructs more number of images in Fig. 13 compared to the flood fill-based method. However, it fails for images 2, 4, 10 and 13 where there are not enough external forces at the gap region to pull the snake towards the concavity of the character. For images 4 and 10 in Fig. 13, the active contour-based method fails to reconstruct as it gives white patches for those characters. On the other hand, the proposed method detects the gaps in all images and the reconstructed characters are recognized correctly except for the last character. For the last character in Fig. 13, the method fills the gap but it does not preserve the shape of the characters. Therefore, the OCR engine fails to recognize it. Note that the symbol ‘ ‘ in Fig. 13 indicates that the OCR engines return an empty string for the input character where OCR engine fails to recognizes. Hence the proposed method outperforms the existing methods in terms of both visual results and recognition results.

4.2. Qualitative results on reconstructed images

To measure the quality of the reconstruction results of the proposed and existing methods (prior to recognition), we use relative error. We create synthetic images corresponding to characters in our dataset to find the error between the synthetic data and the input data (broken characters) and between the synthetic data and the reconstructed data (output). The relative error is computed using the number of contours and gaps in the synthetic, input and output data

Relative error of input:

$$E_{input} = \sum_i \frac{\left|cg_{input\ i} - cg_{synthetic\ i}\right|}{max\left(cg_{input\ i}, cg_{synthetic\ i}\right)} \tag{5}$$

Relative error of output:

$$E_{output} = \sum_i \frac{\left|cg_{output\ i} - cg_{synthetic\ \ i}\right|}{max\left(cg_{output\ i}, cg_{synthetic\ i}\right)} \tag{6}$$

where $cg\_synthetic_i$, cg_input_i and cg_output_i are the total numbers of contours and gaps of the ith synthetic image, the ith input image and the ith output image, respectively.

Table 1 shows the total number of contours and gaps of synthetic, input and output data. It is observed from Table 1 that the number of contours and gaps of the synthetic and output data are almost the same while there is a huge difference between those of the synthetic and input data. As a result, it is concluded that the proposed method reconstructs broken characters properly. It can be seen in Table 2 where the relative error is high for the input data while those of the output data are almost zero. Hence, the proposed method is useful in reconstruction of broken characters to increase the recognition rate of video characters.

Table 1
Number of contours and gaps for qualitative measures.

Parameters	Synthetic	Input	Output
Total contour (TC)	209	339	217
Total gap (TG)	0	303	0
Measure (=TC+TG)	209	642	217

Table 2
Quality measures for character reconstruction.

Measures	Input	Output
Relative error	0.67	0.03

4.3. Recognition results on character images

In this section, we consider character recognition rate (CRR) to evaluate whether the reconstructed results have preserved shape of the actual character or not. If the OCR engine recognizes the reconstructed character correctly then we consider that the shape is preserved. Otherwise it can be concluded that the reconstruction algorithm does not preserve the shape of the character while filling in the gaps in the contours. In this work, the horizontal and the vertical filling algorithms as well as the small gap filling algorithms are the key steps for character reconstruction because the horizontal and the vertical filling algorithms can fill in gap without small gap filling algorithm and vice versa. Therefore, to find contribution of each step in reconstruction, we evaluate the above two steps independently in terms of character recognition rate. The character recognition rates for the horizontal and the vertical filling algorithms and the small gap filling algorithm are reported in Table 3. In order to show improvement in recognition rate, we compute CRR for both the steps before reconstruction and after reconstruction (Table 3). It is observed from Table 3 that the horizontal and the vertical filling algorithms give slightly lower results compared to the result of small gap filling algorithm. This is because the horizontal and the vertical filling algorithms do not involve the *Large gap filling algorithm* and the *Border gap filling algorithm* steps for character reconstruction. On the other hand, the small gap filling algorithm fills gaps based on end points regardless of gap lengths and shapes. Therefore, these two algorithms complement each other to achieve a better accuracy as reported in Table 3. Note that the CRR before reconstruction (26%) are the same for all the three methods because the inputs are same set of broken characters. It is observed from Table 4 that the performances after reconstruction of the two existing methods are lower than that of the proposed method. This is because the flood fill-based method fills the gaps correctly only for small gaps but not for large gaps. In addition, the flood fill-based method fills in not only the character pixels but also the background pixels (even with better stopping criteria). Similarly, the active contour-based method can only handle small gaps. It is mentioned in [22] that when there is a gap, the lack of external forces at that region will cause the snake to go inside the character instead of stopping at the character

Table 3
Character recognition rate for the horizontal and vertical, and small gap filling algorithms separately.

Steps	**Character recognition rate (%)**
Horizontal and vertical filling algorithm	41.3
Small gap filling algorithm	44.1
Proposed method (combined)	71.2

Table 4
Character recognition rate and improvements of the proposed and existing methods.

Method	**Before reconstruction (%)**	**After reconstruction (%)**	**Improvements (%)**	**Time (s)**
Flood fill-based method [18]	26.0	20.3	−5.7	1.08
Active contour-based method [22]		54.9	28.9	28.3
Proposed method		**71.2**	**45.2**	**1.98**

contour. For small gaps, this problem can be overcome by setting appropriate snake tension and rigidity parameter values. However, it still fails for big gaps. There are two other drawbacks of the active contour-based method: (i) it requires ideal images (or templates) to guide the compensation of external forces at the gap regions and (ii) a good initial contour is required for the snake to converge to the correct contour of the character. For these reasons, the two existing methods give poor results for our dataset. On the other hand, the proposed method works well for both small and large gaps and thus achieves the best performance in our experiment. Character recognition rate of the proposed method is promising, given that the problem is challenging and it does not require any extra information like templates. Table 4 shows that the processing time per image of the proposed method is slightly longer than the flood fill-based method but much shorter than the active contour based method. This is because the flood fill-based method takes the Canny output of the original image as input for reconstruction, while the active contour based method involves expensive computations and it requires more iterations to meet its convergent criteria. For our method, the major computational time is spent in connected component analysis to find end points and to eliminate blobs, branches and loops for reconstruction. This computation time is still much shorter than that in the active contour based method. However, the processing time depends on the platform of the system and the data structure of the program. It is confirmed from the improvement in terms of CRR reported in Table 4 that the proposed method outperforms the existing methods with reasonable computation time.

4.4. Reconstruction on general objects

This section shows that the proposed method can be used for general object contour reconstruction if the objects possess symmetry. Object contour is an important feature for many computer vision applications such as general object recognition [9,28,29], industrial object recognition [30], face and gesture recognition [31]. However, due to reasons such as poor resolution and low contrast (e.g. when dealing with video images) and occlusion, it is not always possible to extract the complete contour of an object. Such cases lead to partial information about the contour, i.e. for some regions there is an uncertainty about whether they are part of the contour or part of the background. If it is the former case, reconstruction is required to get the complete contour. Otherwise, many shape-dependent computer vision methods (e.g. those mentioned above) will degrade in performance.

Fig. 14 shows the reconstruction result of the proposed method for a general object. It is clear that different kinds of gaps are recovered: horizontal/vertical gaps, gaps on the inner/outer contour while the outer/inner contour is preserved, and gaps where both contours are broken but the nearby regions are preserved. Hence, it is possible to extend the proposed method to general objects.

5. Conclusion and future work

We have proposed a novel method for reconstructing contours of broken characters in video images. To the best of our knowledge, this is the first attempt to increase the recognition rate by proposing the novel RRT. The normalized absolute gradient feature and the Canny edge map are used to extract the initial character contours from the input gray image. RRT helps to identify medial pixels, which are used to fill in the horizontal, vertical gaps, large gaps and border gaps. Finally, the remaining small gaps are reconstructed based on the mutual nearest neighbor concept. Experimental results in terms of qualitative and quantitative measures show that the proposed method outperforms the existing methods. However, the CRR of the proposed method is lower than the rate that achieved on high resolution camera based character images because the proposed method does not preserve the shapes of the characters while filling the gaps.

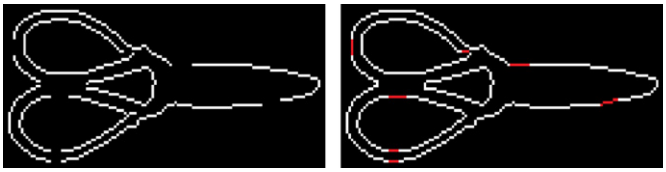


Fig. 14. Reconstruction result of the proposed method for a general object (scissors).

In future, we plan to extend the reconstruction algorithm so that it preserves the shapes of the characters to improve the CRR. We would also like to explore reconstructing general objects, including non-symmetric objects, using the proposed RRT. One possible way is to relax the condition in Section 3.2.3 (for confirming the presence of a symmetric region) to handle small gaps in non-symmetric objects.

Acknowledgment

This work is done jointly by National University of Singapore and Indian Statistical Institute, Kolkata, India. This research is also supported in part by the A*STAR Grant 092 101 0051 (NUS WBS R252-000-402-305). The authors are grateful to the anonymous reviewers for their constructive suggestions to improve the quality of the paper.

References

[1] D. Chen, J. Odobez, Video text recognition using sequential Monte Carlo and error voting methods, Pattern Recognition Letters (2005) 1386–1403.

[2] D. Doermann, J. Liang, H. Li, Progress in camera-based document image analysis. In: Proceedings of the ICDAR, 2003, pp. 606–616.

[3] X. Chen, J. Yang, J. Zhang, A. Waibel, Automatic detection and recognition of signs from natural scenes, IEEE Transactions on Image Processing (2004) 87–99.

[4] P. Zhou, L. Li, C.L. Tan. Character recognition under severe perspective distortion, in: Proceedings of the ICDAR, 2009, pp. 676–680.

[5] Y.F. Pan, X. Hou, C.L. Liu, Text localization in natural scene images based on conditional random field, in Proceedings of the ICDAR, 2009, pp. 6–10.

[6] Y.F. Pan, X. Hou, C.L. Liu., A robust system to detect and localize texts in natural scene images, in: Proceedings of the DAS, 2008, pp. 35–42.

[7] D. Chen, J.M. Odobez, H. Bourlard, Text detection and recognition in images and video frames, Pattern Recognition (2004) 595–608.

[8] S.H. Lee, J.H. Kim, Complementary combination of holistic and component analysis for recognition of low-resolution video character images, Pattern Recognition Letters (2008) 383–391.

[9] A. Ghosh, N. Petkov, Robustness of shape descriptors to incomplete contour representations, IEEE Transactions on PAMI (2005) 1793–1804.

[10] X. Tang, X. Gao, J. Liu, H. Zhang, A spatial-temporal approach for video caption detection and recognition, IEEE Transactions on Neural Networks (2002) 961–971.

[11] C. Wolf, J.M. Jolion, Extraction and recognition of artificial text in multimedia documents, Pattern Analysis and Applications (2003) 309–326.

[12] T. Lelore, F. Bouchara, Document image binarization using Markov field model, in Proceedings of the ICDAR, 2009, pp. 551–555.

[13] C. Wolf, D. Doermann, Binarization of low quality using a Markov random field model, in Proceedings of the ICPR, 2002, pp. 160–163.

[14] C.L. Tan, R. Cao, P. Shen, Restoration of archival documents using a wavelet technique, IEEE Transactions on PAMI (2002) 1399–1404.

[15] Q. Chen, Q.S. Sun, P.A. Heng, D.S. Xia., A double-threshold image binarization method based on edge detector, Pattern Recognition (2008) 1254–1267.

[16] G. Guo, J. Jin, X. Ping, T. Zhang, Automatic video text localization and recognition, in: Proceedings of the International Conference on Image and Graphics, 2007, pp. 484–489.

[17] Z. Saidane, C. Garcia, Robust binarization for video text recognition, in Proceedings of the ICDAR, 2007, pp. 874–879.

[18] Z. Zhou L. Li, C.L. Tan, Edge based binarization for video text images, in: Proceedings of the ICPR, 2010, pp. 133–136.

[19] J. Wang, H. Yan, Mending broken handwriting with a macrostructure analysis method to improve recognition, Pattern Recognition Letters (1999) 855–864.

[20] D. Yu, H. Yan, Reconstruction of broken handwritten digits based on structural morphological features, Pattern Recognition (2001) 235–254.

[21] B. Allier, H. Emptoz, Degraded character image restoration using active contours: a first approach, in: Proceedings of the ACM Symposium on Document Engineering, 2002, pp. 142–148.
[22] B. Allier, N. Bali, H. Emptoz, Automatic accurate broken character restoration for patrimonial documents, International Journal on Document Analysis and Recognition (2006) 246–261.
[23] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proceedings of the CVPR, 2010, pp. 2963–2970.
[24] T.Q. Phan, P. Shivakumara, C.L. Tan, A gradient vector flow-based method for video character segmentation, in: Proceedings of the ICDAR, 2011, pp. 1024–1028.
[25] P. Shivakumara, T.Q. Phan, C.L. Tan, A Laplacian approach to multi-oriented text detection in video, IEEE Transactions on PAMI (2011) 412–419.
[26] J. Park, G. Lee, E. Kim, S. Kim, Automatic detection and recognition of Korean text in outdoor signboard images, Pattern Recognition Letters (2010) 1728–1739.
[27] Tesseract. 〈http://code.google.com/p/tesseract-ocr/〉.
[28] D.S. Guru, H.S. Nagendraswamy, Symbolic representation of two-dimensional shapes, Pattern Recognition Letters (2006) 144–155.
[29] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: Proceedings of the CVPR, 2003, pp. 409–415.
[30] P. Nagabhushan, D.S. Guru, Incremental circle transform and eigenvalue analysis for object recognition: an integrated approach, Pattern Recognition Letters (2000) 989–998.
[31] X. Fan, C. Qi, D. Liang, H. Huang, Probabilistic contour extraction using hierarchical shape representation, in: Proceedings of the ICCV, 2005, pp. 302–308.

P. Shivakumara is a research fellow in the Department of Computer Science, School of Computing, National University of Singapore. He received BSc, MSc, MSc Technology by research and PhD degrees in computer science, respectively, in 1995, 1999, 2001 and 2005 from the University of Mysore, Mysore, Karnataka, India. In addition to this, he has obtained educational degree (BEd) in 1996 from Bangalore University, Bangalore, India.

From 1999 to 2005, he was project associate in the Department of Studies in Computer Science, University of Mysore, where he conducted research on document image analysis, including document image mosaicing, character recognition, skew detection, face detection and face recognition. He worked as a research fellow in the field of image processing and multimedia in the Department of Computer Science, School of Computing, National University of Singapore, from 2005 to 2007. He also worked as a research consultant in Nanyang Technological University, Singapore, for a period of 6 months on image classification in 2007. He has published around 90 research papers in national, international conferences and journals. He has been reviewer for several conferences and journals. His research interests are in the area of image processing, pattern recognition, including text extraction from video, document image processing, biometric applications and automatic writer identification.

Trung Quy Phan is pursuing the graduate degree from the Department of Computer Science, School of Computing, National University of Singapore, Singapore. He is currently a research assistant with the School of Computing, National University of Singapore. His current research interests includes image and video analysis.

Souvik Bhowmick was a researcher in the Department of Computer Science, National University of Singapore. He is currently a research assistant with Indian Statistical Institute, Kolkata, India. His research interests are in image and video analysis.

Chew Lim Tan is a professor in the Department of Computer Science, School of Computing, National University of Singapore. He received his BSc (Hons) degree in physics in 1971 from University of Singapore, his MSc degree in radiation studies in 1973 from University of Surrey, UK, and his PhD degree in computer science in 1986 from University of Virginia, USA. His research interests include document image analysis, text and natural language processing, neural networks and genetic programming. He has published more than 300 research publications in these areas. He is an associate editor of Pattern Recognition, associate editor of Pattern Recognition Letters, an editorial member of the International Journal on Document Analysis and Recognition. He is a member of the Governing Board of the International Association of Pattern Recognition (IAPR). He is also a senior member of IEEE.

Umapada Pal received his PhD from Indian Statistical Institute and his PhD work was on the development of Printed Bangla OCR system. He did his post doctoral research on the segmentation of touching English numerals at INRIA (Institut National de Recherche en Informatique et en Automatique), France. During July 1997–January 1998 he visited GSFForschungszentrum fur Umwelt und Gesundheit GmbH, Germany, to work as a guest scientist in a project on image analysis. From January 1997, he is a faculty member of the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata. His primary research is Digital Document Processing. He has published 160 research papers in various international journals, conference proceedings and edited volumes. In 1995, he received student best paper award from Chennai Chapter of Computer Society of India. He received a merit certificate from Indian Science Congress Association in 1996. Because of his significant impact in thedocument analysis research domain of Indian language, TC10 and TC11 committees of IAPR (International Association for Pattern Recognition) presented 'ICDAR Outstanding Young Researcher Award' to him in 2003. In 2005–2006 he has received JSPS fellowship from Japan government. He has been serving as a program committee member of many conferences including International Conference on Document Analysis and Recognition (ICDAR), International Workshop on Document Image Analysis for Libraries (DIAL), International Workshop on Frontiers of Handwritten Recognition (IWFHR), International Conference on Pattern recognition (ICPR), etc. Also, he is the Asian PCChair for 10th ICDAR to be held at Barcelona, Spain, in 2009. He has served as the guest editor of special issue of VIVEK journal on document image analysis of Indian scripts. Also currently he is coediting a special issue of the Journal of Electronic Letters on Computer Vision and Image Analysis. He is a life member of IUPRAI (Indian unit of IAPR) and senior life member of Computer Society of India.