A New Iterative-Midpoint-Method for Video Character Gap Filling

Palaiahnakote Shivakumara, Ding Bei Hong, Danni Zhao, Chew Lim Tan and Uma Pal

Abstract—We propose a new Iterative-Midpoint-Method (IMM) for video character gap filling based on end pixels and neighbor pixels in the extracted contour of a character. The method obtains the Enhanced Gradient Image (EGI) for the given gray character image to sharpen text pixels. Max-Min clustering and K-means clustering algorithm with $K=2$ are applied on the EGI to obtain text candidates. To clean up the background information, the intersection of the text candidate image and the Sobel of the input image is considered. The method extracts edges in Canny of the input image corresponding to pixel in the intersection results, which we call the potential candidates having possible contour of the character with fewer disconnections. From the contour, we identify the correct pair of end pixels based on mutual nearest neighbor criteria. The three midpoints obtained from the two end pixels and their two preceding pixels are noted. The distance between these three consecutive midpoints is used to predict a new midpoint. From the new midpoint, the method recursively computes midpoints till it reaches end pixels, which results in updated new end pixels. In this way, the method repeats midpoint computation iteratively to fill the complete gap between two end pixels. The method has been tested on 500 images which include 200 character images from video, 200 character images from ICDAR-2003 competition data and 100 images from object data to evaluate the performance. The comparative study shows that the proposed method is superior to a baseline method in terms of recognition rate.

Keywords—Video character recognition, Video document analysis, Character gap filling

I. INTRODUCTION

The unpredictable characteristics of video which causes disconnections, loss of information, loss of shapes of characters make the scene and graphics character recognition problem more complex and challenging [1-4]. There are methods for scene character recognition in camera based images in literature that have achieved so far reasonable accuracy up to 67% [1]. This indicates that achieving good accuracy for video characters is still an elusive goal in the field of pattern recognition and image processing because video frames are much more complex than camera scene images. It is evident that if we apply conventional character recognition methods directly on video characters, the methods give poor recognition rate, typically from 0% to 45% [2]. 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, complex background etc. [5].

Hence, in the present paper, we focus on filling gaps on broken contours of characters to increase the recognition rate because recovering shape or contour is challenging and interesting compared to other problems. The contour gap filling based on end pixels and midpoint computation is motivated by the way the human visual system works. Research has shown that when an object is occluded, human 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 [6]. This kind of filling is useful in many kinds of objects, from real world objects to industrial objects and even organs in the human body.

Video character recognition methods use either enhancement by integrating temporal frames or by improving binarization or by filling gaps in the character contours to improve the 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 [2]. The method [7] uses corner points to identify candidate text regions and proposes color clustering to separate the foreground and background information. However, this method assumes that text lines should have uniform color pixels. For the same purpose, the method [8] proposes a 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 [9] 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 contrast, complex background of video images and distortions even if we give segmented video character as input. There is another method [10] in the document analysis to fill small gaps caused by degradations and distortions in contour to improve character recognition. As the above methods are designed for document images, they rely heavily on Connected Component (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 Iterative-Midpoint-Method (IMM) to fill in the gaps of a broken character based on end pixels and pixels neighbouring the end pixels. The novelty of this method is that it preserves the shape of the character while filling the gaps and it is not confined to characters (fills gaps in general objects) as IMM requires only end pixels and two proceeded pixels from end pixels to fill the gap. Here, we choose two previous pixels for gap filling based on empirical study. In this way the method is effective and is different from that in the literature.

II. PROPOSED METHOD

In this work, we use our method [11] based on gradient vector flow to segment the characters from the text lines. The character segmentation method works well for low contrast text, complex background and different oriented text in video frames. Besides, this method converts non-horizontal text lines to horizontal text lines based on text line direction. Therefore, non-horizontal characters are treated as horizontal characters in this work. Thus, the output of the character segmentation method [11] is considered as input for the proposed method in this work.

The proposed method is divided into three sub-sections. Text candidates from the segmented character image are obtained using
gradient and clustering methods discussed in Section A. From the text candidates, the possible restoration of missing text information by suppressing non-text components with the help of Sobel and Canny edge-image of the input images, which we call potential text candidates, is presented in Section B. In section C, we propose a novel Iterative-Midpoint-Method (IMM) to fill the gaps in the extracted components.

A. Text Candidates Selection

For the example segmented character shown in Figure 1(a), the method obtains the gradient image to sharpen the text pixel because it is a fact that gradient operation gives high values for the text pixel and low values for the non-text pixels as shown in Figure 1(b) where one can see edge information is brightened. Further, to increase the gap between text and non-text pixels, we perform 3×3 sliding window operation on gradient image, which selects the maximum gradient value among 8 neighbors for each pixel in the image. This results in an Enhanced Gradient Image (EGI) as shown in Figure 1(c) where it is noticed that text pixels are brighter than the pixel in Figure 1(b). The reason for performing this operation is that the previous step may enhance high contrast pixels in the background along with text pixels. To reduce this effect, we propose EGI operation on gradient image.

To separate text and non-text pixels from the EGI, we propose a Max-Min clustering algorithm that selects Max and Min gradient values from the EGI and then it classifies the gradient values in EGI into Max-Cluster (text cluster) if the gradient value is closer to the Max gradient value otherwise it classifies it into Min-Cluster (non-text cluster). The output of Max-Cluster can be seen in Figure 1(d) where pixels are classified as text. Since Max-cluster is a simple classification, it may extract non-text pixel as text pixel due to complex background and non-uniform illumination.

Therefore, to avoid this effect, we propose K-means with K=2 on the results of Max-Cluster to identify the text candidates as shown in Figure 1(e) where we can see loss of text information.

B. Potential Text Candidates Selection

As it is observed from the previous section that it is hard to restore the complete text information due to unfavorable characteristics of video, the method intersects Sobel of the input images shown in Figure 2(a) with the text candidate image, which gives fine edge information as shown in Figure 2(b) where it is seen that this criteria eliminates text information due to the problem of low resolution. This is because Sobel gives good response to high contrast information but not for low contrast information. To restore the lost text information from the results of intersection, we use Canny of the input image as Canny gives better results compared to the Sobel for low contrast images, while for the high contrast images, Canny gives lots of erratic edges which may affect the filling process. One such example is shown in Figure 2(c) where edges for background and text are extracted. Therefore, we extract edge components, which we call corresponding to the pixel in intersection result, which we call potential text candidates as shown in Figure 2(d) where the method extracted only text edges. In this way, we get the possible contour of the character. However, we can still see some gaps on the contour in extracted edge components as shown in Figure 2(d) despite of our effort to get the complete contour as discussed in the above. This motivates us to propose a novel method for filling the gaps on the contour to improve the video character recognition rate. The result of iterative midpoint method can be seen in Figure 2(e) for the image shown in Figure 2(d), where the gap is filled completely.

C. Iterative Midpoint Method

For each potential text candidate image shown in Figure 2(d), we propose a novel Iterative-Midpoint-Method (IMM) based on identifying the correct pair of end pixels and the pair of pixels neighboring the end pixels because we believe that the behavior of the pixels in the gap (between two end pixels) is same as the behavior of the pair of end pixels and the pair of their neighbor pixels. In other words, we try to interpolate the behavior of the pair of end pixels and the pair of neighbor pixels to fill the gap between the two end pixels. It is inspired from the work presented in [12] where the midpoint computation from the end pixels is used for identifying small gaps at the corner of the contour of the general objects. This idea is good for small curve gaps and single gaps on the contours of the objects but not for multiple gaps and large gaps that are common in case of video characters as it uses angle information of the segments to fill the gaps. The method identifies the correct pair of end pixel first based on mutual nearest neighbor criteria to avoid making wrong pair of end pixels that may exist due to noise in the background or end pixels of other gaps. Let N_n be the neighbor pixel of the contour pixel P(x,y). The correct end pixels of P(x,y) are identified as defined in equation (1).

\[
\text{P_n0} \text{ and P_n1 are the two end pixels given by the equation (1). If P_n0 is the nearest neighbor according to Euclidean distance to P_n0 and if P_n0 is the nearest neighbor to P_n0 then the P_n0 and P_n0 are said to be satisfy the mutual nearest neighbor criteria and hence P_n0 and P_n0 are considered as the correct pair of end pixels of the contour as shown in Figure 3(a) to fill the gap between them.}
\]

To interpolate the behavior of the pair of end pixels and the pair of neighbor pixels, the method computes midpoints, say P_m0, P_mid, P_mid1, P_mid2 for the pair of end pixels and two pairs of previous pixels pair found from the end pixels as in the equations (2)-(4), respectively, where P_m0, P_mid, P_mid1, P_mid2 denote the pair of end pixels, P_m0, P_m0 denote first previous pixels pair from the end pixel and P_m0, P_m0 denote second previous pixel pair from the end pixels. The distance between (P_m0, P_mid) and (P_mid, P_mid1) is estimated to find larger distance, say d_max as in the equation (5). The larger distance is used to predict a new midpoint (P_mid_predict) as defined in the equation (6) as shown in Figure 3(a). The method recursively computes the midpoint between a new predicted midpoint and the pair of end pixels until it reaches the end pixels (P_m0, P_m0), which predicts new contour pixels in the gap by adding one pixel to each end pixel and newly added pixel are considered as new end pixels for next iteration. This process repeats iteratively until the gap is closed as shown in Figure 3(b) where the gap is closing after a few iterations. The final filled contour
can be seen in Figure 3(c) where it is noticed that the gap is filled completely according to shape of the character in 8th iteration. Hence, our iterative midpoint method not only fills gaps but also preserves the shape. This is the advantage of the proposed method.

\[
\text{End Pixel } P(x, y) = \begin{cases} 
\text{true,} & N_{g} = 1 \\
\text{false,} & N_{g} \neq 1
\end{cases}
\]

\[
P_{\text{mid}0} = P_{n0} - \frac{d(P_{n0} - P_{n1})}{2}
\]

where \(P_{n0} > P_{m0}\)

\[
P_{\text{mid}1} = P_{n1} - \frac{d(P_{n1} - P_{m1})}{2}
\]

where \(P_{n1} > P_{m1}\)

\[
P_{\text{mid}2} = P_{n2} - \frac{d(P_{n2} - P_{m2})}{2}
\]

where \(P_{n2} > P_{m2}\)

\[
d_{\text{max}} = \max\{d(P_{\text{mid}_0} - P_{\text{mid}_1}), d(P_{\text{mid}_1} - P_{\text{mid}_2}), d(P_{\text{mid}_0} - P_{\text{mid}_2})\}
\]

\[
P_{\text{mid\_predict}} = P_{\text{mid}0} + (\frac{P_{\text{mid}0} - P_{\text{mid}2}}{d_{\text{max}}}) * d_{\text{max}}
\]

\(\vec{n}(P_{\text{mid0}, P_{\text{mid}2}})\): Direction vector from \(P_{\text{mid0}}\) to \(P_{\text{mid}2}\)

III. EXPERIMENTAL RESULTS

We consider the video data, ICDAR 2003 competition data and COIL-20 object data for the purpose of experimentation. Since there is no standard database for filling gap of video characters, we create our own data which includes a diverse set of data chosen from news video, sports video and movie video. The method is evaluated on 200 video character images, 200 ICDAR 2003 competition character images chosen randomly and 100 COIL-20 object data. In total, 500 data are considered. We use relative error (\(E_{\text{input}}\) and \(E_{\text{output}}\)) for measuring the quality of reconstruction as defined in equation (7) and (8) and the recognition rate to test whether the filled character shape is preserved or not with the available Google tessreact OCR [13]. To compute the relative error, we create synthetic images using Paint as reference data and Canny of the synthetic images are considered for calculating the number of gaps and the contours. Similarly for the filled data given by the proposed method, we calculate the number of contours and the gaps. In the equation (7) and (8), \(cg_{\text{input}}, cg_{\text{synthetic}}, cg_{\text{output}}\) denote number of contours and gaps in input data, synthetic data and the output data, respectively.

We implement a recent binarization method [9] as a baseline for comparison. This baseline method uses a modified flood fill algorithm to improve video character recognition rate by filling small gaps on the contours.

\[
E_{\text{input}} = \sum \max(\text{cg}_{\text{input}}, \text{cg}_{\text{synthetic}}) \quad (7)
\]

\[
E_{\text{output}} = \sum \max(\text{cg}_{\text{output}}, \text{cg}_{\text{synthetic}}) \quad (8)
\]

A. Experimental Results on Video Data

Sample results of the proposed method and the baseline method for filling gaps at different position on contour of the character are shown in Figure 4 where the proposed method fills gaps correctly as shown in (b) and the shape is also preserved as shown in (c) while the baseline method fails to fill the gaps as shown in (d). It is also observed from Figure 4 that all filled characters by the proposed are recognized correctly by the OCR engine except the last character due to shape problem whereas characters filled by the baseline methods are not recognized correctly. In Figure 4, the recognition characters are shown within quote (‘’). Blank ‘ ‘ indicates that OCR returns none. The qualitative and quantitative measures of the proposed method and the baseline method on video data are reported in Table 1 where the relative error for the input is higher than the output and the output error is almost zero. Therefore, the proposed method fills gaps well for the characters. Table 1 shows that the proposed method gives better recognition rate compared to the baseline method as we can see large improvements (Impr) in recognition rate compared to before and after filling while the baseline method gives negative improvements due to inherent problems of the method. Therefore, the proposed method outperforms the baseline method in terms of recognition rate.

\[\text{Figure 4. Sample results on video data. (a) Canny image with gaps, (b) Gap filled image by the proposed method, (c) Flood filled image and (d) Flood filled image using existing method. Recognition results of the respective characters are shown in the right side of (c) and (d).} \]

Table 1. Qualitative and quantitative measures on video data

<table>
<thead>
<tr>
<th>Method</th>
<th>Relative error</th>
<th>Methods</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output</td>
<td>Before</td>
<td>After</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.81</td>
<td>0.10</td>
<td>Proposed 47.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flood fill</td>
</tr>
</tbody>
</table>

B. Experimental Results on ICDAR 2003 Data

Though the aim of this work is to fill gaps in video characters, we also test the proposed method on camera based scene characters to show the effectiveness of the method. Sample results of the proposed method and the baseline method are shown in Figure 5 where we can observe that the proposed method fills gaps well compared to the baseline method. The results of second row in Figure 5 show the proposed method is capable of removing background edges without affecting the shape of the character but the baseline method is not so. For the results of last row shown in Figure 5, though the proposed method fills the gap clearly, the OCR recognizes “g” as “8” because the shape of the character look like shape of “8”. It is also noticed from Figure 5 that the character filled by the proposed method are recognized by the OCR engine correctly except for the last character. On the other hand, for the characters filled by the baseline method, OCR fails to recognize all characters except “R”. Table 2 shows the relative error of the input is higher than the output and the output error is almost zero. This infers that the proposed method works well for scene characters also. Table 2 shows that the recognition rate of the proposed method is higher than the baseline method and there is large improvement over recognition rate before filling because the proposed method has the ability to fill the gaps according to character shape.

From Table 1 and Table 2 it can be observed that the recognition rate for the scene character is higher than the video characters before filling. This shows that scene character have good shape with small gaps due to high contrast compared to video characters. However, the improvement in recognition rate for the scene characters is lower than video characters. The
reason for this may be that ICDAR dataset includes only scene characters with high contrast and complex background while video dataset includes both caption and scene characters with low contrast and complex background. Thus more disconnections and large gaps are available in video data but the same is not true for scene characters. Since the proposed method is good in filling large gaps and disconnections, it shows more improved recognition rate for the video data compared to scene data.

![Sample results on ICDAR data](image)

Figure 5. Sample results on ICDAR data. (a) Canny image with gaps (b) Gap filled image by the proposed method (c) Flood filled images and (d) Flood filled image using baseline method. Recognition results of the respective characters are shown in the right side of (c) and (d)

Table 2. Qualitative and quantitative measures on ICDAR data

<table>
<thead>
<tr>
<th>Method</th>
<th>Relative error</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output</td>
<td>Before</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.6</td>
<td>0.03</td>
</tr>
<tr>
<td>Flood fill</td>
<td>59.0</td>
<td>58.0</td>
</tr>
</tbody>
</table>

![Iterative midpoint method for 3D COIL objects](image)

Figure 6. Iterative midpoint method for 3D COIL objects

C. Experimental Results on Object Data

Since the proposed method uses end pixel and preceded pixels from the end pixel to fill the gap on the contour of the character, the method can be applied on objects to fill the contour gap if any. It is noted from the methods [14, 15] that due to occlusion, the disconnections and loss of objects shape are a common problem that reduces the accuracy of object recognition. Therefore, to show the generic property of the proposed method, we test on COIL-20 3D object database [6] and the proposed method is evaluated in terms of relative error as reported in Table 3. Sample results of the proposed method is shown in Figure 6 where (a) is an input object image, (b) is canny of the input images having gap and (c) is the final filled-up result. Table 3 shows that the relative error of the input is higher than the relative error of the output. This concludes that the proposed method fills gaps correctly even for objects. Hence the proposed method is generic and it can be used for filling gaps in objects.

Table 3. Qualitative measure on COIL-20 object data

<table>
<thead>
<tr>
<th>Methods</th>
<th>Relative error</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>0.97</td>
<td>0.12</td>
</tr>
</tbody>
</table>

IV. Conclusion

In this work, we propose a novel method for filling gaps in video characters, camera based character and general objects based on end pixels and midpoint computation. The method extracts character contour using Max-Min clustering and K-means clustering algorithms. The Sobel and Canny of the input images are used for restoring the potential text candidates. The iterative-midpoint-method is proposed to fill the gaps in potential text candidate images. Experimental results shows that the proposed method reconstructs well as relative error is low for the output and the proposed method outperforms the baseline method in terms of recognition rate on video and camera based characters.

Acknowledgment

This work done jointly by National University of Singapore (NUS) and Indian Statistical Institute (ISI), Kolkata, India and this research is supported in part by A*STAR grant 092 101 0051 (WBS no. R252-000-402-305).

REFERENCES