Automatic splicing of Chinese single-sided shreds based on character feature and typesetting characteristics

Hongrong Yang and Yunqiong Wang*

School of Information Science and Technology, Yunnan Normal University Kunming, 650500, China

*Corresponding author’s e-mail: wangyunqiong@ynnu.edu.cn

Abstract. In this paper we proposed an automatic splicing algorithm for regular cross-cut single-sided shreds of pure Chinese document, which was based on the characteristics of character and typesetting. The algorithm began with extracting the characteristics including character width, character height, line height and line spacing, after which the shreds meeting the typesetting requirement were spliced according to the edge similarity. Afterwards, the horizontal projection vectors of the shreds were corrected and the shreds were clustered by row according to horizontal projection similarity. A greedy algorithm with rejection strategy was then used to complete the intra-cluster horizontal splicing of shreds based on edge similarity. The experiment results revealed that the algorithm proposed in this paper had the characteristics of both high splicing speed and high accuracy.

1. Introduction
The automatic splicing technique of document shreds is a typical application in the field of image processing and pattern recognition. At present, it is mainly applied in the aspects of historical document restoration, judicial material evidence restoration and military intelligence acquisition. The shreds splicing algorithms mainly depend on contour-based method or content-based method. On the one hand, contour-based method is appropriate for the conditions where apparent key points exist along the shreds contour. He Pengfei [1] put forward a local splicing method based on contour characteristics and a global matching algorithm based on ACO algorithm. Meanwhile, Ying Shan [2] proposed a fragment matching method based on corner curvature which required huge computations. Moreover, F. Amigoni [3] published a two-dimensional fragment splicing method based on multi-scale analysis. Additionally, Hori [4] proposed a method for segmentation of fragment contour curves according to feature points. On the other hand, the content-based method is applied in the majority of regular shreds splicing algorithms. Zhao Bo [5] presented a splicing algorithm based on the information quantity, which could realize the splicing of 9*9 shreds with an accuracy of 55%, but the algorithm was unable to splice shreds with blank margin. Zixiao Pan [6] proposed a clustering algorithm based on ant colony algorithm. Nevertheless, there were misjudgments during the clustering process and manual intervention was required. In terms of the splicing restoration of 11*19 shreds, Yin Yuping [7] proposed an automatic splicing algorithm with the application of dynamic clustering. The accuracy of splicing was 94.3%, and 12 pairs of shreds were manually intervened. In addition, Fu Guanghui [8] proposed an automatic splicing method grounded on clustering algorithm and ant colony algorithm, and the accuracy of splicing was 95.1%. Shuxuan Guo [9] put forward a splicing algorithm based on row clustering, error evaluation function, optimal reconstruction route searching and human mediation, of which the accuracy was 98.5%. Besides, Junhua Chen [10] proposed an algorithm depending on shred
clustering by line height and line spacing and intra-cluster splicing by ant colony optimization algorithm, and the accuracy of splicing was 95%.

In this paper we put forward a greedy algorithm with rejection strategy based on the similarity of cut edges, which achieved automatic splicing of 11*19 cross-cut shreds, and the accuracy reached 99.57%. The algorithm started with feature extraction of both character and typesetting. Afterwards, the rejection strategy was adopted to achieve vertical and horizontal splicing of the shreds of which the typesetting characteristics conformed requirement and the cut edges were located inside the character. Then the horizontal projection vectors of shreds were corrected according to the splicing result and the shreds pieces were clustered by row based on the horizontal projection similarity. Ultimately, the edge similarity of the shreds was used to achieve intra-cluster horizontal splicing by means of the greedy algorithm.

2. Feature extraction
As far as Chinese document shreds with the same font size and line spacing were concerned, the character width, character height, line height and line spacing remained unchanged, while incorrect splicing could result in variation of these characteristics. According to these constraints, some improper splicing of shreds could be eliminated to improve splicing efficiency and accuracy.

2.1. Feature extraction of line height and line spacing based on horizontal projection

2.1.1. Horizontal projection of shreds. Within the shred images, the text lines were separated by blank lines. Therefore, the shreds were projected horizontally to distinguish the text lines from the blank lines. The horizontal projection characteristic $H_{\text{horproj}}$ of the n × m shred image $f_i$ was calculated according to Formula (1).

$$H_{\text{horproj}} = \begin{cases} 1, & \text{if } \exists f_i(c, r) < 255, c = 1, \ldots, n; r = 1, \ldots, m \\ 0, & \text{else} \end{cases}$$

In Figure 1, (a), (c), (e), and (g) were shred images, while (b), (d), (f), and (h) were the corresponding horizontal projection result graphics.

![Figure 1. Schematic diagram of horizontal projection of shred images.](image)

Then the 1 run-length value $L_1[t]$ and 0 run-length value $L_0[t]$ of $H_{\text{horproj}}$ of all shred images were analyzed and used to calculate the line height and line spacing. Figure 2 revealed the statistical result of experiment material.

![Figure 2. Statistic result of horizontal projection run-length.](image)
2.1.2. Feature extraction of line height and line spacing. Ideally, the corresponding \( t \) value of the non-zero maximum of the shreds line height distribution function \( L_1[t] \) was the line height \( LH \) and the corresponding \( t \) value of the non-zero minimum of the line spacing distribution function \( L_0[t] \) was the line spacing \( LS \).

\[
LH = \arg \max_{L_1[t] > 0} (t) \\
LS = \arg \min_{L_0[t] > 0} (t)
\] (2)

The top-bottom separated character in the document would cause \( LS \) to be far smaller than the actual value. With the intention of obtaining a stable \( LS \), the distribution function \( LSH[t] \) calculated the sum value of the top-bottom adjacent 1 run-length value of text line and 0 run-length value of blank line. The LSH diagram was shown in Figure 3. Affected by characters composed of top (bottom) horizontal stroke or top-bottom separated strokes, text indentation, and segment tailer, \( LSH[t] \) approximately subordinated to the normal distribution with the mean value \( LH+LS \). The statistical results of \( LH+LS \) values obtained from experimental material were presented in Figure 4.

![Figure 3. The LHS feature diagram.](image)

![Figure 4. Statistical results of LSH value.](image)

The top-bottom separated characters would reduce the value of \( LH+LS \), while the text indentation and the segment tailer would enlarge the value of \( LH+LS \). On the purpose of eliminating the influence of these factors, \( LH+LS \) values which were too small or too large in Figure 4 could be removed according to the "3\( \sigma \)" rule of the normal distribution. The corrected LSH results were shown in Figure 5.
According to the property of the normal distribution, the mean value of the corrected $LHS$ distribution in Figure 4 was $LH+LS$. Combining the result of Formula (3), a stable $LS$ value could be obtained.

### 2.2. Feature extraction of character width based on vertical projection of text line

It was similar to the feature extraction method of line height that the vertical projection characteristics $Verpro_i$ of the text line $j$ of shred image $i$ was obtained by Formula (4):

$$Verpro_{ij} = \begin{cases} 1, & \text{if } \exists f_i(c, r) < 255, r = s_j, \ldots, e_j; c = 1, \ldots, k \\ 0, & \text{else} \end{cases} \quad (4)$$

Where $s_j$ and $e_j$ were the starting line number and ending line number of the text line $j$ in the shred image $i$.

The corresponding $t$ value of non-zero maximum of $Verpro$'s one run-length statistical value $Y_{ij}[t]$ was the character width feature $CW$.

$$CW = \arg\max_{Y_{ij}[t] > 0} (t) \quad (5)$$

### 3. Vertical splicing of shreds

When the horizontal cut was inside the text line, the information of strokes located on top and bottom sides of the cut edges could be used to constrain vertical splicing and reduce the amount of alternative shreds. The successfully spliced shreds could provide more accurate information of text line location and horizontal projection characteristics, thereby improving the accuracy of subsequent row clustering.

#### 3.1. Binarization of shred images

To facilitate the calculation of cut-edge similarity, the shred image $f_i$ was binarized by Formula (6) to obtain the corresponding binary image $g_i$.

$$g_i(x, y) = \begin{cases} 0, & \text{if } f_i(x, y) = 255 \\ 1, & \text{else} \end{cases} \quad (6)$$

#### 3.2. Vertical splicing of shreds based on edge similarity

When the horizontal cut was inside the text, the sum of the character height along the top and bottom cut edges was not exceeding $LH$, while the sum of the character width was not exceeding $CW$. For shreds that met these two constraints, vertical splicing was performed in accordance with the similarity of strokes on both sides of the cut (Algorithm 1). The main steps were as follows:

Step 1 Similarity calculation: The similarity function $SDU_j(x)$ between the upper shred $i$ and the any lower shred $x$ was defined as:

![Figure 5. Statistical diagram of corrected LHS.](image)
\[ SDU_i(x) = n - \sum_{k=1}^{n} |g_i(k, m) - g_i(k, 1)| \]

Step 2 Initializing the incorrectly spliced shreds set \( V_i = \{ \} \).

Step 3 Rejection strategy: Let \( UH_i \) be the inferior character height of the upper shred \( i \), \( DH_j \) be the superior character height of the lower shred \( j \), \( CW_i \) be the character width of the upper shred \( i \) and \( CW_j \) be the character width of the lower shred \( j \). If \( UH_i + DH_j \leq LH \) or \( CW_i + CW_j \leq CW \), then \( j \) is added to \( V_i \) and \( SDU_i = 0 \).

Step 4 Vertical splicing: For shred \( i \) and \( j \), if \( j = \arg \max \{ SDU_i(x) \} \), then shred \( i \) and \( j \) were successfully spliced. Add \( i \) to \( V_j \) and \( SDU_i = 0 \).

3.3. Correction of horizontal projection according to vertical splicing and typesetting characteristics

The characters with top (bottom) horizontal strokes and missing text lines could reduce the locating accuracy of \( \text{Horpro} \) characteristics on the actual position of the text line, which might lead to subsequent clustering errors.

If the vertical splicing of the shreds had been successful, the \( \text{Horpro} \) characteristics were corrected according to the Formula (7) for the vertically merged shreds.

\[
\text{Horpro}_{ij} = \begin{cases} 
1 & \text{if } \text{Horpro}_{ij} = 1 \\
\text{Horpro}_{ij} & \text{else}
\end{cases} \quad (7)
\]

Figure 6 showed part of the shreds with corrected horizontal projection characteristics.

Figure 6. Schematic diagram of horizontal projection correction.

In Figure 7 (a) and (b), (c) and (d), (e) and (f) belonged to the same clustering group respectively, but (b), (d), (f) were all classified incorrectly into a clustering group before horizontal projection correction. When Formula (7) was used to correct the horizontal projection characteristic, the wrong clustering result could be avoided.

Figure 7. Shreds with missing text lines.

4. Horizontal splicing of shreds based on similarity of cut strokes

When the vertical cut was inside the character, the stroke information on the left and right sides of the cut could be used as splicing constraints, which meant that the sum of the character width on bilateral sides was not exceeding \( CW \). In addition, the horizontal projection of spliced shreds met the typesetting requirements. These two constraints could be applied to eliminate most of the incorrect splicing among the shreds. Furthermore, the number of alternative shreds could be reduced and the horizontal projection could be corrected. To guarantee correct splicing, the calculation of similarity was constrained by the consistency of the stroke direction based on the characteristics that gray value in the center of the stroke was small while that of the marginal area was large (Algorithm 2). The main steps were as follows:
Step 1 Similarity calculation: When the vertical cut was inside the character, the gray-value gradient directions of the stroke on the edge of the shreds were the same or the opposite. The gray-value gradient direction $\alpha_i(x, y)$ of the shred image $f_i$ at the position of $(x, y)$ was obtained by Formula (8).

$$\alpha_i(x, y) = \arctan \left[ \frac{g_y}{g_x} \right] \quad (8)$$

Where, $g_x = \frac{\delta f_i(x, y)}{\delta x} = f_i(x + 1, y) - f_i(x, y), g_y = \frac{\delta f_i(x, y)}{\delta y} = f_i(x, y + 1) - f_i(x, y)$.

The gray-value gradient directions could be aggregated into 8 categories including $-\frac{3\pi}{4}, -\frac{\pi}{2}, -\frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{3\pi}{4}$ and $\pi$, which were represented by 0 to 7 respectively. The similarity $SLR_i(j)$ between the right edge of the shred $i$ and the left edge of the shred $j$ was obtained by Formula (9).

$$SLR_i(j) = \begin{cases} SLR_i(j) + 1 & \text{if } \alpha_i - \alpha_j = 0 \text{ or } |\alpha_i - \alpha_j| = 4 \\ 0 & \text{else} \end{cases} \quad (9)$$

Similarly, the similarity between the left side of the shred $j$ and the right side of the shred $i$ $SRL_j(i)$ was calculated.

Step 2: Rejection strategy. According to the typesetting characteristics of the text, the incorrectly spliced shreds array $F_{ij}$ was calculated by Formula (10).

$$F_{ij} = \begin{cases} 1 & \text{if } RCW_{ik} + LCW_{jk} > CW \\ 0 & \text{else} \end{cases} \quad (10)$$

Where $RCW_{ik}$ and $LCW_{jk}$ were the character width on the right side and left side of the cut respectively.

Step 3 Horizontal splicing: For shred $i$ and shred $j$, if $j = \operatorname{argmax}(SLR_i(x))$ and $i = \operatorname{argmax}(SRL_j(x))$, then $i$ and $j$ were spliced successfully. For other shred $k$, $SLR_i(k) = 0, SRL_j(k) = 0$.

5. Row clustering of shreds according to the similarity of horizontal projection characteristics

The splicing complexity elevated exponentially with the increase in the number of shreds. By converting the splicing problem of numerous shreds into several subproblems, the complexity of splicing could be reduced and the splicing efficiency could be improved. The shreds with the same transverse cut had consistent or similar $Horpro$ characteristics. As shown in Figure 8, (a) and (b), (c) and (d), (e) and (f) belonged to the same cluster groups respectively.

![Figure 8](image_url)

Figure 8. Schematic diagram of shreds in the same cluster groups.

For $p$ rows $\times$ $q$ columns cross-cut shreds, they were divided into $p$ clustering groups containing $q$ shreds within each division based on $Horpro$ similarity (Algorithm 3). The main steps were as follows:

Step 1: Calculating the horizontal projection similarity $Similarity_{ij}$ between the right edge of the left shred $i$ and the left edge of the right shred $j$ within a same clustering group by Formula (11), and initializing the set $V_i = \{i\}$.

$$Similarity_{ij} = \begin{cases} Similarity_{ij} + 1 & \text{if } Horpro_{ik} = Horpro_{jk} \\ Similarity_{ij} & \text{else} \end{cases} \quad (11)$$

Step 2: Sorting the similarity of each shred with any other shred in descending order. When the $Similarity_{ij}$ ranked among the top $q$-1 of the similarity belonging to shred $i$ with any other shred,
and the $\text{Similarity}_{ji}$ ranked among the top $q-1$ of the similarity belonging to shred $j$ with any other shred, then $i$ and $j$ were categorized to the same cluster group. $j$ was added to the set $V_j$, and $i$ was added to the set $V_i$;

Step 3: When $V_i = V_j$ and $|V_i| = q$, shred $i$ and shred $j$ were divided into a same cluster group;

Step 4: The similarity value between the clustered shreds and the non-clustered shreds were modified to 0;

Step 5: Repeating steps 2 to 4 for the non-clustered shreds until all of them were clustered.

6. Vertical sorting among clustering groups and intra-cluster horizontal splicing

After accomplishment of shred clustering, horizontal projection features of all shreds in the same clustering group were merged. Combining the vertical splicing results according to section 2.2 along with the merged projection features, the algorithm 1 was applied for vertical sorting among the clustering groups.

For all the unspliced shreds within the same clustering group, the algorithm 2 was used to achieve intra-cluster splicing referring to the shreds which had been spliced vertically and horizontally.

7. Experimental results and analysis

To verify the efficacy of the algorithm proposed in this paper, 10 pages of Chinese A4 documents were selected as experimental materials. Each page of the document was cut into 11 rows*19 columns to produce 209 shreds with a resolution of 72*180 for experiments. The operating environment was configured as intel Core i5-7200U, dual-core CPU 2.5GHz, and memory 8G.

Table 1. Comparison of splicing accuracy of different splicing methods.

| Material | with rejection strategy | without rejection strategy | with rejection strategy | without rejection strategy |
|----------|-------------------------|----------------------------|------------------------|-----------------------------|
| Material 1 | 100% | 90.4% | Material 6 | 99.04% | 79.43% |
| Material 2 | 100% | 91.8% | Material 7 | 98.09% | 73.68% |
| Material 3 | 99% | 80% | Material 8 | 100% | 78.47% |
| Material 4 | 100% | 81.8% | Material 9 | 100% | 67.94% |
| Material 5 | 100% | 86.1% | Material 10 | 100% | 89.95% |

Table 1 demonstrated that the accuracy of the splicing algorithm with the rejection strategy was significantly higher than that of the algorithm without rejection strategy. The average accuracy of 10 experiment materials was 99.57%. Due to the row clustering and the greedy algorithm, the algorithm had high splicing speed, and the average running time was within 20 seconds.

The experimental results also revealed that completely correct splicing was not achieved in 3 experimental materials. The reason was that there were multiple pairs of shreds where vertical cuts located in the blank column within the same cluster group, as shown in Figure 9.

Figure 9. Schematic diagram of mis-spliced shreds.

Figure 10 revealed the results of automatic splicing and restoration of document shreds of 3
experimental materials.

Figure 10. Splicing results of experimental materials.

8. Conclusion

An automatic splicing algorithm for single-sided document shreds based on the character feature and typesetting characteristics was proposed in this paper. This algorithm is suitable for the shreds of pure Chinese documents with the same font and font size. However, the shreds in this experiment were obtained by computer-simulated cutting. Therefore, for real cutting shreds or torn fragments, the cut edge might be deformed, and the accuracy of splicing could be affected.

Acknowledgments

Each step of this paper, from topic selection to completion, was guided by the teacher Yunqiong Wang. My supervisor devoted all of her energies. I would like to give my highest respect and thanks to my supervisor!

References

[1] He Pengfei. (2009) Shreds Splicing Based on Ant Colony Optimization Algorithm. National University of Defense Technology, Hunan.
[2] S. Shan. (2002) New Measurements and Corner-guidance for Curve Matching with Probabilistic Relaxation. International Journal of Computer Vision, 46(2):157-171.
[3] De Gama Leitão H C, Stolfi J. (2002) A multiscale method for the reassembly of two-dimensional fragmented objects, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(9): 1239-1251.
[4] Hori K, Imai M, Ogasawara T. (1999) Joint detection for potsherds of broken earthenware, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2: 440-445.
[5] Zhao Bo, Zhou Yu, Zhang Zhengyu, etc. (2014) Automatic Splicing Restoration of Shredded Chinese Documents Based on Information Richness. Journal of Computer Aided Design and Graphics, 27(6): 1039-1046.
[6] Pan Z, Wang M. (2017) A new method of shredded paper image restoration based on ant colony algorithm, In: 2017 Chinese Automation Congress (CAC).
[7] Yin Yuping, Liu Wanjun, Zhang Chong, Liu Yongchao. (2014) Automatic Document Splicing Algorithm Based on Dynamic Clustering. Computer Engineering and Applications, 50(18): 162-166+170.
[8] Fu Guanghui, Hua Yun, Chen Junhua, Pan Xinwei. (2019) A Cross-Cutting Shredded Paper Restoration Algorithm Based on Clustering and Ant Colony Algorithm. Mathematical Practice and Knowledge, 49(15): 199-209.
[9] Shuxuan Guo, Songyang Lao, Jinlin Guo.(2015) A Semi-automatic Solution Archive for cross-cut shredded text documents reconstruction. ICIG 2015, Part I, LNCS 9217, pp. 447–461.

[10] Junhua Chen, Daguan Ke, Zhanghong Wang, Youjun Liu.(2018) A high splicing accuracy solution to reconstruction of cross-cut shredded text document problem. Multimed Tools Appl. 77:19281–19300.