A Semi-automatic Solution Archive for Cross-Cut Shredded Text Documents Reconstruction

Shuxuan Guo[1(✉)], Songyang Lao[1], Jinlin Guo[1], and Hang Xiang[2]

[1] National University of Defense Technology,
Science and Technology on Information Systems Engineering Laboratory,
Changsha, People's Republic of China
gsxuan6688@163.com, laosongyang@vip.sina.com, gjlin99@nudt.edu.cn
[2] Chongqing Key Laboratory of Emergency Communication,
Chongqing, People's Republic of China
feixiang788@gmail.com

Abstract. Automatic reconstruction of cross-cut shredded text documents (RCCSTD) is important in some areas and it is still a highly challenging problem so far. In this work, we propose a novel semi-automatic reconstruction solution archive for RCCSTD. This solution archive consists of five components, namely preprocessing, row clustering, error evaluation function (EEF), optimal reconstructing route searching and human mediation (HM). Specifically, a row clustering algorithm based on signal correlation coefficient and cross-correlation sequence, and an improved EEF based on gradient vector is separately evaluated by combining with HM and without HM. Experimental results show that row clustering is effective for identifying and grouping shreds belonging to a same row of text documents. The EEF proposed in this work improves the precision and produces high performance in RCCSTD regardless of using HM or not. Overall, extra HM boosts both of the performance of row clustering and shred reconstructing.

Keywords: RCCSTD · Row clustering · Signal correlation coefficient · Cross-correlation sequence · Gradient vector · Human mediation

1 Introduction

Reconstruction of shredded documents is important in some applications, such as recovering the evidence in criminal investigation, repairing historical documents in archeology and obtaining intelligence in military.

Traditionally, reconstruction of shredded text documents is conducted manually, which is difficult and time-consuming, especially for large amount of documents or shredded pieces. With the development of computer technology, many semi-automatic or automatic reconstruction technologies of shredded documents have been proposed to improve the efficiency and precision. However, it is still far from a perfect-solved problem.

Y.-J. Zhang (Ed.): ICIG 2015, Part I, LNCS 9217, pp. 447–461, 2015.
DOI: 10.1007/978-3-319-21978-3_39

In this work, we focus on the reconstruction of cross-cut shredded text documents (RCCSTD). Here, we define:

Definition 1. *Cross-cut Shredding*
All shreds of cross-cut shredded text documents are rectangular with the same width and height, and the size of shreds is smaller than the original documents which are printed only one side.

In Fig. 1, we illustrate a cross-cut shredding with 5×5 cut pattern.

Supposed that the output of a shredding device is a set of shreds, denoted as $S = \{s_0, s_1, s_2, ..., s_{n-1}\}$, where n is the number of shreds. Each shred has some texts and they have the same size of width and height. Let s_n be the virtual shred, which is a blank piece of paper with the same size as the ones in S. It could be safely ignored because there is no information on it [1].

A solution to the RCCSTD problem can be represented by an injective mapping $\Pi = S \rightarrow D^2$, that is, each shred is mapped to a position (x, y) in the Euclidean space, where $x, y \in D = \{0, 1, 2, ..., n\}$. The remaining positions of the whole document are filled with many copies of virtual shreds s_n [1].

The reconstruction could be divided into three subproblems, namely preprocessing, shred matching and optimal reconstructing route searching. Given all shreds from one page of a text document, in the preprocessing, all shreds are scanned into images and transformed into "ideal formal shreds" to support the following steps. Then, an EEF is used to measure the matching degree between shreds. Finally, terminative reconstruction are conducted by an optimal route search strategy.

In this paper, a novel document reconstruction system is presented with a novel row clustering algorithm, an improved EEF combining with an effective search strategy and some necessary HM which are used to balance the speed and

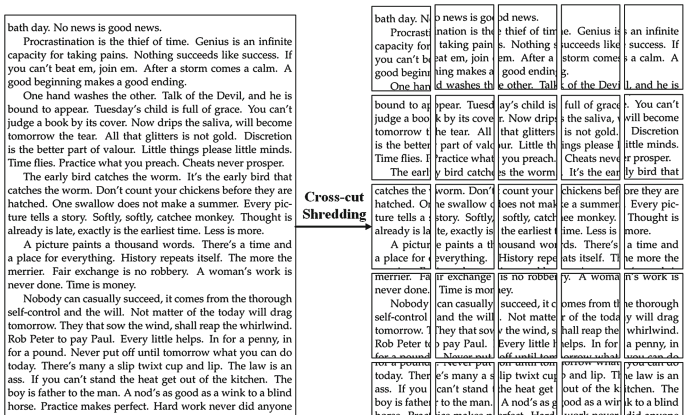

Fig. 1. Illustration of cross-cut shredding

precision during the row clustering and shred reconstructing. Our contribution in this work are that (1) we propose a novel row clustering algorithm and an improved EEF, furthermore, (2) we investigate and evaluate the efficiency and effectiveness of HM in RCCSTD.

The reminder of this paper is organized as follows: related work on reconstruction of shredded documents is briefly reviewed in Sect. 2. Details of the novel semi-automatic solution archive are described in Sect. 3. Section 4 presents the experiments for evaluating the techniques proposed in this work and Sect. 5 concludes this work.

2 Related Work

The text document reconstruction can be categorized into different kinds of problems, including jigsaw puzzles [2,5], strip shredded documents reconstruction [6,9,11], manually torn documents reconstruction [3,4] and the cross-cut shredded documents reconstruction [1,10,13,14].

Strip shredding is illustrated in Fig. 2. The reconstruction of strip shredded text documents (RSSTD) problem can be directly transformed to the known travelling salesman problem (TSP) [7], which is NP-complete. There is no algorithm that solves every instance of this problem in polynomial time. Furthermore, the RSSTD is a special RCCSTD problem without vertical or horizontal cut, since the RCCSTD is NP-hard, which means that it is at least as complex as RSSTD. Therefore, much effort has been made on finding out efficient optimal reconstructing route searching.

In [15], Ukovich et al. used MPEG-7 descriptors in the context of strip shredded documents, while in [10], Prandtstetter et al. presented various algorithms to solve the RCCSTD problem like an ant colony optimisation, a variable neighbourhood search. In contrast, little progress has been made in developing the

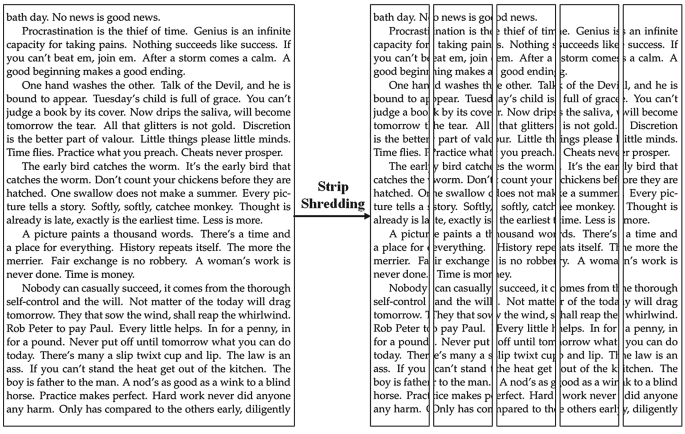

Fig. 2. Illustration of strip shredding

score function or the error evaluation function. In [1] Biesinger et al. provided a formal definition of EEF. It selects a cost based on the weighted difference of the adjacent pixels on either side of candidate matching pairs. Some recent work has begun to use characters as features for pairwise piece matching. A system for shreds matching using recognition of partial characters was developed by Perl et al. in [8].

Human mediation (HM) is less used in previous papers. Prandtstetter and Raidl in [9] took advantage of user mediations while they used a variable neighborhood search approach for strip shredded documents reconstruction. In some situations, the HM is proved to be very helpful for RCCSTD problem.

3 Semi-automatic Solution Archive for RCCSTD

In this section, we give a detailed introduction to the novel semi-automatic solution archive for RCCSTD proposed in this work. The framework consists of five specific components, namely, preprocessing, row clustering, EEF model, optimal reconstructing route searching and HM. The flowchart is shown in Fig. 3.

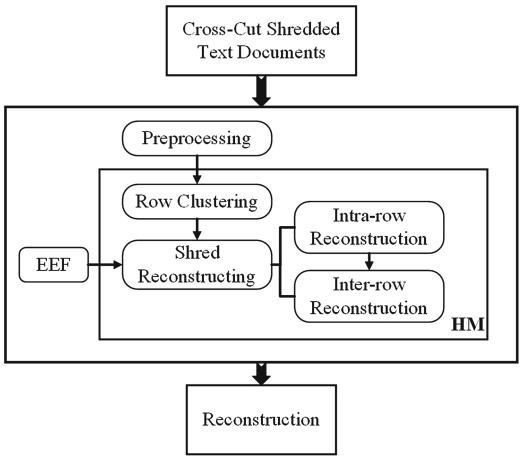

Fig. 3. The flowchart of semi-automatic aolution archive for RCCSTD

Note that the HM is used both in row clustering and shreds reconstructing.

3.1 Preprocessing

Initially, cross-cut shredded text documents are scanned and transformed into digital images, which could be perspective corrected photos, scans, or synthetic images without background and overlapping. After obtaining the images of all shreds, we label each shred image with a unique number. Finally, 8 bits gray scale matrices representing shreds images are taken as input for reconstruction system.

3.2 Row Clustering

RCCSTD owns NP-hard complexity. Therefore, identifying and grouping shreds of the same original row of text documents by clustering algorithm is helpful for reducing computation time and human mediations. Here, we propose a clustering algorithm based on signal coefficient, which significantly decreases the calculating and matching times in optimal reconstructing route searching.

There are at least four baselines in each row of the text documents, and these baselines divide the characters into three cells. This is called four-line ruled printing format, which constraints the distribution of texts in a row strictly. An example of four-line ruled serif characters is shown in Fig. 4:

Fig. 4. Illustration of the four-line ruled printing format

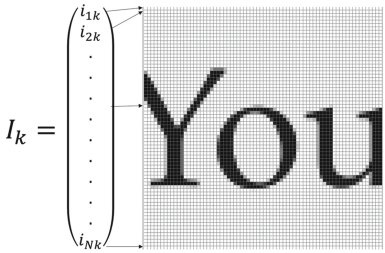

Fig. 5. Illustration of the pixel projection

Hence, there is a certain correlation between the shreds coming from the same row due to the distribution constraint of texts. Firstly, we need to calculate pixel projection of each shred by using the inner and edge information of shreds. The pixel projection vector of image k is donated as $I_k = (i_{1k}, i_{2k}, ..., i_{Nk})^T$, where N is the number of rows of pixel images and i_{nk} represents the amount of non-zero pixels in row n. Figure 5 illustrates the pixel projection. Here, all projection vectors of shreds are considered as signals, and based on the signal correlation coefficient and cross-correlation sequence, we propose the row clustering algorithm. This algorithm is inspired by the discovery in our preliminary experiments that the peaks of the cross-correlation sequence of shreds from the same row appear at almost the same lag pixel, while the ones from different rows have apparent positional deviation of lag pixel.

Supposed that I_i, I_j denotes pixel projection vectors of two shreds i, j respectively, the correlation coefficient is calculated by:

$$c_{ij} = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} \left(\frac{I_i(k) - \bar{I}_i}{\sigma_{I_i}} \right) \left(\frac{I_j(k) - \bar{I}_j}{\sigma_{I_j}} \right) \tag{1}$$

where $\bar{I}_i, \bar{I}_j$ is the mean of I_i, I_j, and $\sigma_{I_i}, \sigma_{I_j}$ is the standard deviation of I_i, I_j. c_{ij} measures not only whether I_i and I_j are in a line or not, but also the linear

relation between them. The higher value of c_{ij} indicates the stronger relation between I_i and I_j.

The cross-correlation sequence is calculated by:

$$R_{ij}(m) = E\{I_i(n+m)I_j^*(n)\} = E\{I_i(n)I_j^*(n-m)\} \tag{2}$$

where, $R_{ij}(m)$ represents cross-correlation sequence, $E\{*\}$ is the expected value operator. I_j^* denotes the conjugate vector of I_j. The cross-correlation sequence is obtained in order to further determine row groups.

In Algorithm 1 pseudo code is given for row clustering method based on signal coefficient.

Algorithm 1. Row Clustering based on Signal Coefficient

```
1  Input: Shreds initialization: pixel projections set I
2  for pixel projection of each shred
       calculate the correlation coefficient with other shreds
3    Sim(i, j) ← coCoefficient(I_i, I_j)
4  end
5  Determine the cluster centers using K-means Algorithm
6  r ← 1
7  while r ≤ RowNumber do
8    Randomly select a cluster center shred i
9    Rank coCoefficients between each shred and cluster center shred i
10   j ← 2
11   while Sim(i, j) ≥ threshold α
12     Accusim(i, k) ← Xcorr(I_i, I_j)
13     Calculate the difference between lag pixel positions of peaks
14     If difference ≤ β
15       Cluster shred j into Row i
16     elseif Call for HM module
17     end
18     k ← k + 1
19     j ← j + 1
20   end
21   r ← r + 1
22 end
23 Output: Clustered Rows
```

This algorithm can be divided into two steps.

- Step 1 (Line 1 to Line 9): Clustering shreds roughly. Firstly, correlation coefficients between each pair of shreds are calculated. Then we determine cluster centers by using K-means clustering method. Afterwards, we choose one of the cluster center randomly and rank the correlation coefficients based on their relation to the cluster center in descending order.
- Step 2 (Line 10 to Line 22): Clustering shreds finely. Based on results from Step 1, we choose a threshold α to determine whether the shred i needs to

calculate the cross-correlation sequence against the cluster center. The cross-correlation sequence needs to be calculated between cluster center shred i and the shreds with correlation coefficient greater than threshold α. When the differences between the peak lag pixel positions less than or equal to β, the shreds are in the same row with cluster center shred i, otherwise, the HM module is employed.

Repeat Step 1 and Step 2 for each cluster center. Shreds that are not clustered into any groups are categorized by means of human mediation. Finally, the algorithm clusters the shreds into several row groups.

3.3 Error Evaluation Function Based on Gradient Vector

EEF is the key to RCCSTD problem. However, it attracts little interest recently. Here, we propose an error evaluation function (EEF) based on gradient vector of edges. Compared with the EFF described in [10] and the cost function in [1], the EEF proposed in this work focuses on both the relationship between the pixels of edges and the diversification of the gray scales from edge to edge.

Supposed that there are two shreds, represented by s_u, s_v, the pixel value of left edge and right edge of which are denoted as LE_{s_u}, RE_{s_u}, LE_{s_v} and RE_{s_v} respectively. For shred s, the gradient at the position (x, y) is calculated by:

$$grad(s) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \end{bmatrix} \tag{3}$$

The gradient angle is:

$$\theta(x, y) = arrctan(\frac{g_x}{g_y}) \tag{4}$$

This angle represents the direction of maximum changing rate at (x, y). Therefore, we define the EEF, expressed as follows:

$$c(u, v) = \sum_{i=1}^{N} \left|(RE_{s_u}(i) - LE_{s_v}(i))\right| \times \left|cos(\theta_u(i) - \theta_v(i))\right| \tag{5}$$

EEF measures not only the difference of the aligned pixels, but also the change direction of aligned pixel along the edge of s_u and s_v. If those two shreds match together perfectly, the EEF between them would yield value 0, while lager values of EEF indicate the dissimilarity between the neighboring shreds.

3.4 Optimal Reconstructing Route Searching

Intra-row Route Searching. According to the definition of EEF described in Eq. 5, the values of EFF between each pair of shreds in the same row group can be calculated. Afterwards, the RCCSTD problem can be described as a TSP model:

- Original node: we use an extra blank shred as the original node.
- Nodes: the shreds need to be reconstructed as well as the blank shreds represents the nodes in graphic model.
- Edges and the costs of edges: Edges are the links between every pair of shreds and the costs of edges are values of EEF between starting nodes and ending nodes of links.

TSP aims at finding out the shortest possible route that visits each node exactly once and returns to the original node. This problem can be transformed into an integer linear problem (ILP), that is:

$$min \sum_{(i,j)\in S} c(i,j)x_{ij} \tag{6}$$

s.t.

$$\begin{cases} \sum_{j\in S} x_{ij} = 1 \\ \sum_{i\in S} x_{ij} = 1 \\ x_{ij} + x_{ji} \leq 1 \\ x_{ii} = 0 \\ x_{ij} \in 0, 1, i, j \in S \\ Only\ One\ Loop \end{cases} \tag{7}$$

In this ILP, the objective function aims to minimizing the total matching cost. The constrains ensure that the solution is a loop without any circle which travels all the nodes with the minimum cost. x_{ij} is a binary variable, and it is set as 1 if the right edge of shred i is matching with the left edge of shred j, otherwise it is set as 0 if the right edge of shred i is not matching with the left edge of shred j. Finally, the original node is removed.

We utilize the branch and bound algorithm, which is represented in [12], for solving this problem. In some special situations, HM is participated in to achieve better performance, and we will introduce it in next subsection.

Inter-row Route Searching. The input for this step is the reconstructed row groups. We use the same optimal reconstructing route searching method as that in intra-row reconstruction, but the left and right edges of shreds need to be replaced with top and bottom edges of rows.

3.5 Human Mediation

In order to address the false shreds match during the whole reconstruction, human mediation module is introduced. It provides better intermediate results to the following steps, especially, when available information of certain shreds is too limited for classifying or reconstructing them automatically. In optimal reconstructing route searching, HM is used to optimize the distance matrix when the value of EEF is beyond the threshold. If participants deem that a pair of shreds are matching, the distance value in matrix will change to 0, besides, the

order of the visit will be limited from the left shred to the right shred. For evaluating the efficiency of HM, we adopt HM times, which is the number of human mediations.

Note that in our experiments, we choose experienced participants in HM module for row clustering and shred reconstructing, since their experience and knowledge may reduce misjudgements.

4 Experiments and Results

In this section, we experimentally evaluate the performance of the proposed methods in this work and compare it with system without using HM by 5 cross-cut shredded documents.

Firstly, in order to obtain the experimental data, we scan 5 different page instances with size of 1368×1980 pixels chosen from 5 different text documents respectively. Instance p001 is from a regular one-column English paper, while instance p002 is a page with table of content. The third instance p003 is a regular one-column English paper but with additional head and footer. Instance p004 is full of references. The last instance p005 is a page with Chinese characters, which is used to examine the applicability of the proposed method. Then, all pages are transformed into formatted gray-scale images. Afterwards, the images are shredded with 5 cutting patterns, namely 5×5, 7×7, 9×9, 11×11, 11×19.

We use the precision, recall and HM times to evaluate effectiveness and efficiency of the proposed method. The precision is calculated by:

$$Precision = \frac{N_T}{N_T + N_F} \tag{8}$$

The recall is computed by:

$$Recall = \frac{N_T}{N_{Total}} \tag{9}$$

where, N_T denotes the number of correct row clustering or matching shreds, while N_F denotes the number of incorrect row clustering or matching shreds. N_{Total} is the total number of shreds required to be clustered and reconstructed.

For HM, we design a user interface, snapshots of which are shown in Fig. 6. This dialog box provides an interface between computer and humans. When the cross correlation sequences with small difference or the EEF is smaller than a threshold, the dialog box is employed. Users can input either 1 or 0 in the dialog box, representing acceptance and rejection respectively.

In HM, we choose users who have much experiences in RCCSTD. Our pre-experiments show that experienced users in HM module can improve the performance of row clustering and shred reconstructing.

Figures 7 and 8 show two examples of acceptance and rejection in row clustering respectively. The HM process works in the same way in optimal reconstructing route searching.

Fig. 6. HM user interface

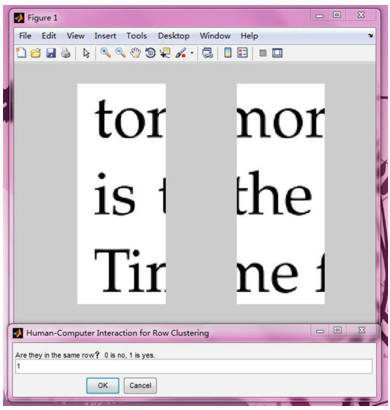

Fig. 7. Example of acceptance

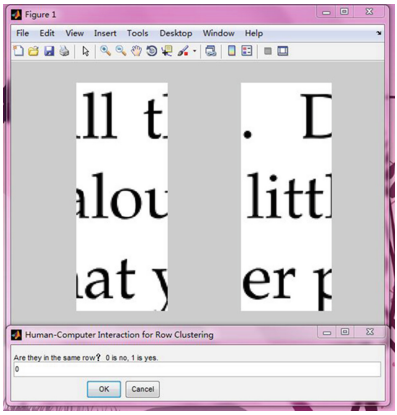

Fig. 8. Example of rejection

4.1 Row Clustering

As the flowchart shown in Fig. 3, after preprocessing, the row clustering algorithm is implemented. Here, we empirically set threshold α as a value in $[0.05, 0.2]$, and threshold β as 1 in Algorithm 1 to trigger the HM module. For demonstration, we give an example, where s_1, s_2, s_3, s_4, s_5 are shreds from 11×11 cutting pattern of instance p001, and s_1, s_2, s_3, s_4 are from a same row, while shred s_5 is from a different row. Cross-correlation sequences between shreds and a cluster center are shown in Fig. 9.

As shown in Fig. 9, cross-correlation sequences of shreds of a same row change nearly simultaneously and reach peaks at nearly the same lag pixel. While cross-correlation sequences of shreds of a different row shows significant differences at the lag pixel where peaks appear.

Since in the clustering, initial cluster centers are selected randomly, and this has impact on the final clustering results, we run each pattern of all 5 instances 10 times and report the average results here. The results are listed in the left part of Table 2.

Our first observation is that for any instance, the precisions and recalls of row clustering decrease overall as it is shredded into more pieces. Even when using method without HM, the precisions and recalls of most cutting patterns of instance p001, p003, p004 and p005 are over 95 %, we deem that this method is

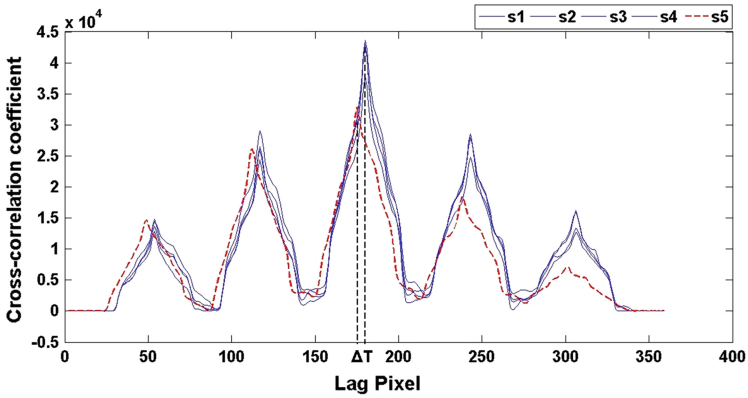

Fig. 9. The cross correlation sequence of 5 shreds of cross-cut documents. Shreds here are from 11×11 cutting pattern of instance p001.

very effective. Furthermore, in some cases, with the assistant of HM, the results of row clustering gains not only marginally. For instance p002, the precisions and recalls are improved significantly when introducing HM, and best results are from with HM by using the K-means cluster centers. Experiments on instance p005 show notable precisions and recalls, which means that the row clustering method is applicable to Chinese documents. In short, the row clustering algorithm proposed in this work is effective for identifying and grouping shreds belonging to a same row of text documents.

Secondly, we observe the HM times. For each instance, it needs more HM as the number of cutting in x- and y- axes increases. This is obvious since the information in each shred is getting less and the clustering needs more human mediations. Especially, for each cutting pattern of instance p002, HM is more needed, this is because instance p002 only contains a table of content, in which, characters are very scanty and the space between two neighboring line is much more wider than that of other instances.

Finally, we compare the performance of random cluster centers with K-means cluster centers. For small or medium cutting patterns of instance p001, p003, p004 and p005, the row clustering using random cluster centers performs a bit better. However, For large cutting patterns and scanty instance p002, row clustering using K-means cluster centers produce stable and better performance. We deem the reasons are that (1) for scanty instances, when selecting cluster centers randomly, it is apt to choose those shreds that contains less row information; (2) initial cluster centers obtained by K-means clustering are more representative and efficient for grouping large number of shreds.

Above all, row clustering algorithm proposed in this work provide three solutions corresponding to different instance respectively. Specifically, row clustering using random cluster centers is more appropriate for small and medium cutting patterns on regular instances, and row clustering using K-means cluster centers

works better on large cutting patterns and scanty instances, while row clustering with HM boosts the performance of the algorithm.

4.2 Shred Reconstructing

In this part, we experimentally test the performance of the proposed EEF in shred reconstructing combined with branch and band algorithm in [12]. Specifically, two methods, EEF with HM and without HM are evaluated. In addition, two methods proposed in previous work, namely method using difference of edge pixels [1] and method using weighted difference (W-Difference) of edge pixels [1], are also evaluated for comparison. Reconstruction precisions of cutting pattern 11×19 are listed in Table 1. As shown in Table 1, the two methods based on the EEF proposed in this work improve significantly the reconstruction performance over all instances, and method with HM outperforms other methods. Especially, for instance p002, the precision of method with HM gains nearly 20 % than that of other three methods.

Table 1. Reconstruction precision using different EEFs. The cutting pattern here is 11×19.

Instance	With HM	Without HM	Difference	W-Difference
$p001$	**96.17 %**	94.74 %	89.47 %	93.30 %
$p002$	**75.60 %**	58.37 %	55.50 %	54.55 %
$p003$	**93.30 %**	92.34 %	90.91 %	91.87 %
$p004$	**94.74 %**	91.87 %	89.00 %	89.95 %
$p005$	**100 %**	98.56 %	97.13 %	97.61 %

More details about all cutting patterns are given in the Shred Reconstructing part of Table 2, including precisions and HM times. The inputs of shred reconstructing module are accurately-clustered shreds groups. Firstly, we observe that on all instance except p002, the reconstruction precisions of small and medium cutting patterns achieve perfectness, 100 %, and even for large cutting patterns, the performance could reach more than 90 %. Meanwhile, with the number of shreds increasing, the reconstruction precisions decline. For instance p002, method Without HM reports modest precisions, but when introducing HM, the construction precision are improved notably. Moreover, shred reconstructing by using extra HM exhibits better performance than method Without HM for any cutting pattern on any instance.

With regard to HM times, experimental results show that there is no need for HM in small and medium cutting patterns reconstruction on p001, p003, p004 and p005, and the HM times in reconstruction increases as a page is cut into more shreds.

Overall, we can conclude that the two methods combined with the EEF proposed in this work improve the precisions and they are effective for RCCSTD.

Table 2. Precision, recall and HM times

Instance	x	y	Row Clustering									Shred Reconstructing		
			Without HM (random)		With HM (random)			With HM (K-means)			Without HM	With HM	
			precision	recall	times	precision	recall	times	precision	recall	precision	times	precision
p001	5	5	95.37%	94.00%	**0**	**100%**	**100%**	9.4	91.76%	89.21%	100%	0	100%
	7	7	**100%**	**100%**	**0**	**100%**	**100%**	21.5	97.53%	96.12%	100%	0	100%
	9	9	96.64%	89.81%	**14.4**	97.53%	**96.71%**	72.6	**97.91%**	96.30%	100%	0	100%
	11	11	**100%**	99.45%	**11**	**100%**	**100%**	79.3	98.34%	97.52%	100%	0	100%
	11	19	**100%**	96.49 %	**51.8**	99.84%	**99.36%**	138.4	99.04%	99.03%	94.74%	33	**96.17%**
p002	5	5	80.87%	67.20%	4	86.86%	84.00%	**3.6**	**93.82%**	**92.00%**	100%	0	100%
	7	7	95.73%	75.92%	30.2	92.90%	89.80%	**23.5**	**96.86%**	**94.39%**	81.63%	51	**87.76%**
	9	9	70.98%	56.34%	**33.4**	70.35%	67.49%	35.7	**86.53%**	**83.95%**	64.20%	193	**86.42%**
	11	11	77.17%	55.37%	117.5	74.37%	69.42%	**104.3**	**83.05%**	**82.47%**	52.07%	279	**86.78%**
	11	19	80.28%	51.52%	330.7	79.14%	70.81%	**152.3**	**82.14%**	**80.67%**	58.37%	418	**75.60%**
p003	5	5	**100%**	**100%**	**0**	**100%**	**100%**	5.7	98.62%	97.26%	100%	0	100%
	7	7	**100%**	95.10%	**0.4**	93.81%	93.20%	12.5	96.93%	**95.16%**	100%	0	100%
	9	9	**100%**	**100%**	**0**	**100%**	**100%**	22.6	93.61%	92.03%	100%	0	100%
	11	11	**99.44%**	98.07%	**28.6**	99.17%	**98.35%**	45.3	94.42%	93.32%	96.69%	10	**100%**
	11	19	**99.31%**	95.37%	**40.3**	98.23%	**97.61%**	69.6	94.74%	92.32%	92.34%	49	**93.30%**
p004	5	5	**100%**	**100 %**	**0**	**100%**	**100%**	3.7	99.32%	96.63%	100%	0	100%
	7	7	**100%**	**100 %**	**0**	**100%**	**100%**	16.3	97.47%	96.57%	100%	0	100%
	9	9	**99.54%**	95.47%	**1.2**	99.17%	**98.77%**	37.9	95.46%	94.33%	100%	0	100%
	11	11	**99.07%**	**95.04%**	**45.5**	92.02%	90.91%	64.3	92.23%	91.26%	100%	0	100%
	11	19	**98.81%**	91.07%	147.3	97.37%	88.76%	**89.4**	93.21%	**92.32%**	91.87%	22	**94.74%**
p005	5	5	**100%**	**100%**	**0**	**100%**	**100%**	6.2	95.21%	94.42%	100%	0	100%
	7	7	**100%**	**100%**	**0**	**100%**	**100%**	15.5	93.36%	92.22%	100%	0	100%
	9	9	**100%**	96.71%	37.8	97.51%	**97.12%**	**33.5**	94.51%	93.23%	100%	0	100%
	11	11	**95.53%**	79.06%	69.5	92.38%	90.50%	**40.9**	92.76%	**92.02%**	98.34%	21	**100%**
	11	19	86.74%	66.03%	323.2	82.83%	76.47%	**59.8**	**95.16%**	**89.35%**	98.56%	56	**100%**

4.3 Summary

In our experiments above, on some instances, high performance could be obtained automatically by using method Without HM over some small even medium cutting patterns. Experimental results also show that HM is effective both in row clustering and shreds reconstructing and the methods using HM is usually able to produce better results than methods without HM. For some instances, extra HM reports much better results, while for some instances, using HM needs much extra overhead, and only marginal performance gains. Considering the complexity of RCCSTD, most HM times is acceptable, since it significantly reduces human labor. Moreover, the successful reconstruction of instance p005 indicates that the solution archive is available for RCCSTD with Chinese characters.

5 Conclusion

In this work, we present a novel semi-automatic solution archive for RCCSTD problem, which consists of five specific components of preprocessing, row clustering based on signal correlation coefficient and cross-correlation sequence, EEF based on gradient vector, optimal reconstructing route searching strategy based on TSP and some necessary Human Mediation.

The row clustering algorithm and shred reconstructing is evaluated by combining with HM and without HM on 5 instances from different text documents.

Through the experimental results of the 5 cutting patterns of each instance, we can figure out that the row clustering algorithm is effective for identifying and grouping shreds belonging to a same row of text documents, and the EEF proposed in this work produces high performance in RCCSTD regardless of using HM. Overall, extra HM boosts both of the performance of row clustering and shred reconstructing. Future work will focus on improving the framework and applying it to reconstruct scanned images on actual shreds from various text documents.

Acknowledgments. The research is partly supported by National Science Foundation of Hunan, China 14*JJ*3010. The authors would like to show gratitude to the tutors and the participants.

References

1. Biesinger, B.: Enhancing an evolutionary algorithm with a solution archive to reconstruct cross cut shredded text documents, na (2012)
2. Chung, M.G., Fleck, M., Forsyth, D.: Jigsaw puzzle solver using shape and color. In: 1998 Fourth International Conference on Signal Processing Proceedings, 1998, ICSP 1998, vol. 2, pp. 877–880 (1998)
3. De Smet, P.: Reconstruction of ripped-up documents using fragment stack analysis procedures. Forensic Sci. Int. **176**(2), 124–136 (2008)
4. De Smet, P.: Semi-automatic forensic reconstruction of ripped-up documents. In: 10th International Conference on Document Analysis and Recognition, 2009, ICDAR 2009, pp. 703–707. IEEE (2009)
5. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. In: Proceedings of the Eighteenth Annual Symposium on Computational Geometry, pp. 82–87. ACM (2002)
6. Justino, E., Oliveira, L.S., Freitas, C.: Reconstructing shredded documents through feature matching. Forensic Sci. Int. **160**(2), 140–147 (2006)
7. Lawler, E.L., Lenstra, J.K., Kan, A.R., Shmoys, D.B.: The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimisation. Wiley, Chichester (1985)
8. Perl, J., Diem, M., Kleber, F., Sablatnig, R.: Strip shredded document reconstruction using optical character recognition (2011)
9. Prandtstetter, M., Raidl, G.R.: Combining forces to reconstruct strip shredded text documents. In: Blesa, M.J., Blum, C., Cotta, C., Fernández, A.J., Gallardo, J.E., Roli, A., Sampels, M. (eds.) HM 2008. LNCS, vol. 5296, pp. 175–189. Springer, Heidelberg (2008)
10. Prandtstetter, M., Raidl, G.R.: Meta-heuristics for reconstructing cross cut shredded text documents. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 349–356. ACM (2009)
11. Ranca, R.: A modular framework for the automatic reconstruction of shredded documents. In: AAAI (Late-Breaking Developments) (2013)
12. Ross, G.T., Soland, R.M.: A branch and bound algorithm for the generalized assignment problem. Math. Program. **8**(1), 91–103 (1975)
13. Schauer, C.: Reconstructing cross-cut shredded documents by means of evolutionary algorithms, na (2010)

14. Schauer, C., Prandtstetter, M., Raidl, G.R.: A memetic algorithm for reconstructing cross-cut shredded text documents. In: Blesa, M.J., Blum, C., Raidl, G., Roli, A., Sampels, M. (eds.) HM 2010. LNCS, vol. 6373, pp. 103–117. Springer, Heidelberg (2010)
15. Ukovich, A., Ramponi, G., Doulaverakis, H., Kompatsiaris, Y., Strintzis, M.: Shredded document reconstruction using MPEG-7 standard descriptors. In: Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004, pp. 334–337. IEEE (2004)