Multimed Tools Appl (2018) 77:19281–19300
https://doi.org/10.1007/s11042-017-5389-z

A high splicing accuracy solution to reconstruction of cross-cut shredded text document problem

Junhua Chen[1,2] · **Daguan Ke**[2] · **Zhanghong Wang**[3] · **Youjun Liu**[1]

Received: 27 February 2017 / Revised: 17 October 2017 / Accepted: 3 November 2017 /
Published online: 13 November 2017

Abstract Reconstruction of Cross-Cut Shredded Text Documents (RCCSTD) plays an important role in both forensics and archeology. It is a special case of the square jigsaw puzzle problem and has attracted the attention of many researchers. In the light of the low accuracy of existing RCCSTD solutions, especially regarding row splicing, this paper proposes a high accuracy splicing solution by using both a combination strategy and a divide-and-conquer strategy. Unlike other approaches based on the Swarm Intelligence Algorithm, where the results and splicing accuracy are bound up with the defined cost function and the number of fragments, in this case a clustering algorithm was used to transform a single RCCSTD problem into several Reconstruction of Strip Shredded Text Document (RSSTD) problems. The dual combination and divide-and-conquer strategies proposed in this paper are designed to improve the splicing accuracy in a row and make the algorithm more stable as the number of fragments in a row increases. Experiments were carried out on 10 text documents (5 Chinese and 5 English), which were shredded into ten patterns. The returned accuracy measures were over 0.95 for the Chinese documents and over 0.85 for the English ones, across all patterns. A comparison is made between our approach and another recently proposed solution, and we conclude that our approach gives both higher splicing accuracy and greater stability regardless of the number of fragments in a row.

Keywords Clustering vector · TSP problem · Ant colony algorithm · Combination strategy · Divide-and-conquer strategy · Reconstruction of cross-cut shredded document problem

✉ Youjun Liu
lyjlma@bjut.edu.cn

[1] College of Life Science and Bioengineering, Beijing University of Technology, No.100 Pingleyuan, Chaoyang District, Beijing 100124, People's Republic of China

[2] School of Information and Engineering, Wenzhou Medical University, Chashan University Town, Wenzhou 325035, People's Republic of China

[3] The 2nd Clinical Medical College, Wenzhou Medical University, Chashan University Town, Wenzhou 325035, People's Republic of China

1 Introduction

Reconstruction of Cross-Cut Shredded Text Documents (RCCSTD) is an important subdomain of forensic science and combinatorial optimization, and it plays a significant role in identification, civil disputes, criminal investigations, and so on [7]. It is usually viewed as a special case of the greedy square jigsaw puzzle problem and has attracted the attention of many researchers [21, 22, 26]. Schauer et al. [26] adopted this view of seeing the RCCSTD problem as a variation upon jigsaw puzzles and defined three kinds of shredded documents: manually torn documents; strip shredded documents; and cross-cut shredded documents (see Fig. 1).

With regard to the reconstruction of manually shredded documents, it is possible to trace more than 50 years of research. In its earliest phase the usual approach was to calculate the degree of the adjacent by using information about a fragments' shape. Wolfson et al. [30] converted the problem of reconstructing manually shredded documents into what is known as the Travelling Salesman Problem (TSP), which focuses on identifying the shortest possible route between a range of points where the pairwise distance is already known. In order to make full use of fragments' shape information, a registration algorithm among shapes is needed, there are some excellent papers focus on this domain [19, 20]. By doing this they were able to bring to bear an already established research technique, which led to a series of achievements [2, 4, 8]. However, as manually torn documents are not the primary focus of this paper, we will not examine this technique further here.

With regard to the Reconstruction of Strip Shredded Text Documents (RSSTD), the content information at the boundary of the fragments is sufficiently good that this is not generally considered to be a difficult problem to solve. Prandtstetter and Raidl [24] solved the RSSTD problem by treating it as a TSP and using a variable neighborhood search approach with a semi-automatic system in the optimization process.

The RSSTD problem is usually considered to be a special case of the RCCSTD problem. Most of the papers about the RCCSTD problem have been published in the past decade. Prandtstetter proved that the RCCSTD problem is a complete Non-Deterministic Polynomial (NP) problem in his thesis [22]. In a later paper, he solved it using ant colony optimization and a variable neighborhood search [23]. Schauer et al. [26] used a genetic algorithm to solve the problem, drawing upon pattern recognition technology. Gong et al. [5] proposed a memetic algorithm based on evolution algorithms, and defined four kinds of operators and a comprehensive cost function, thereby obtaining some satisfactory results. Zhao et al. [33] proposed a new cost function based on Information Quantity to reduce the serious propagation of errors caused by the matching of shreds with low Information Quantity. This approach was particularly well-suited to the reconstruction of Chinese text documents. At the outset, most researchers used an approach that was based upon Swarm Intelligence Algorithms to obtain optimal solutions for the complex spatial configurations that are typical of the kinds of

Fig. 1 Three kinds of shredded documents

problems we are discussing here. This approach is usually viable and can get good results, as the above papers have shown. However, as the number of fragments increases or the boundary information about the fragments decreases, it is much harder to find an optimal solution for complex spatial layouts, and the results can be sensitive to how the cost function is defined. Indeed, it appears to be impossible to define a perfect cost function for the RCCSTD problem.

A solution to this is to cluster fragments into several classes, thereby simplifying the complicated spatial searches. Ukovich et al. [28] used a clustering approach as a preprocessing step before applying the actual reconstruction algorithms. Azzam et al. [27], by contrast, proposed a clustering approach for RCCSTD by defining a cost function as the clustering standard. This gave them high splicing accuracy and an efficient running speed. However, Azzam et al.'s approach proves to be sensitive to the definition of the cost function too. Wang et al. [29] suggested a two-stage approach. Here a clustering algorithm based on text lines was used to cluster fragments into several classes (rows). A memetic algorithm was then used to solve the RSSTD problem. Wang et al.'s paper provides a promising approach to the RCCSTD problem, but its realization in the paper was relatively crude. Xu et al. [31] used a clustering vector to classify fragments into several classes. They then used a genetic algorithm to solve the problem. This appears to be an even more promising approach because it is not sensitive to the cost function and can give high accuracy. However, the method appears to be sensitive to the number of fragments in a row or the number of rows in a shredded text document instead. In our view, the most effective approach to the RCCSTD problem is to include two parts: a feature extraction scheme for clustering fragments; and a heuristic algorithm for the RSSTD problem itself. For the former step, there are numerous papers that focus on feature extraction and learning methods [10, 12–14, 17, 25]. There are also abundant publications relating to the latter step. Papers regarding machine learning approaches that are relevant to the RSSTD problem include [3, 11, 15, 16, 18].

In this paper will be concentrating on the splicing algorithm for rows and how to make the splicing accuracy of the algorithm more stable as the number of fragments in a row increases. For cross-cut shredded documents, our approach is organized as follows: (1) We extract clustering vectors from the fragments according to certain aspects of how words are positioned in the fragment [31]. The clustering vector of the first fragment in each row is defined as the cluster center. (2) We execute a clustering algorithm to classify fragments into several classes based upon the clustering vectors and associated cluster centers. In other words, we convert the RCCSTD problem into an RSSTD problem. This step reduces the need for a quality cost function, with the possibility of similar or higher accuracy, even if the cost function is relatively poor. (3) We define the cost function between the fragments in the same row, transforming the RSSTD problem into a TSP. (4) We use an ant colony algorithm to solve the TSP that has been derived from the RSSTD problem. (5) We use both a combination strategy and a divide-and-conquer strategy to modify the error from the definition of the cost function. This action not only improves the splicing accuracy in a row but also makes the algorithm more stable as the number of fragments in a row increases. (6) As a final step, we merge the fragments between the rows by using the clustering vector extracted from the rows. The accuracy of the final solution is then assessed manually. A flowchart of the whole process is shown in Fig. 2:

The main contribution of this paper is to actualize the feature extraction scheme (vector clustering) first presented in Xu et al. [31] and to make this scheme better suited to the processing of abnormal fragments. Additionally, the paper presents an approach that uses both a combination strategy and a divide-and-conquer strategy to improve splicing accuracy whilst retaining stability as the number of fragments in row increases.

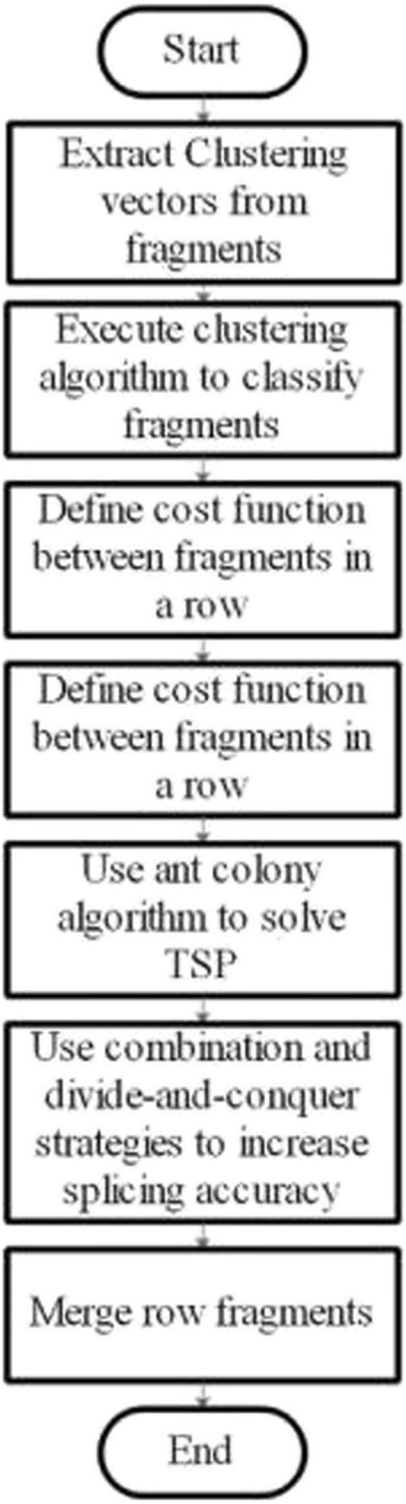


Fig. 2 Flowchart of the whole process

The structure of the paper is as follows: Section 2 introduces our proposed method. In Section 3 we present the results of our experimental testing of the method. Section 4 discusses the results and provide our overall conclusions in Section 5.

2 Method

2.1 Row clustering

2.1.1 The clustering vector

Because the grayscale image matrix data of fragments is very big, we use a clustering vector to describe a fragment. Using feature extraction [31], the image matrix of each fragment can be presented as a 4 × 1 clustering vector. The vector is defined as $CV = [a_1, a_2, a_3, a_4]^T$, where a_1 represents the lower position of an unidentified line word at the top of the fragment and a_4 represents the upper position of an unidentified line word at the very bottom of the fragment. a_2 then represents the upper position of the last identified line word at the bottom of the fragment and a_3 represents the lower position of the last identified line word at the bottom of the fragment (see right of Fig. 3).

The procedures of the feature extraction are showed as follows (see Fig. 3):

P1. If the border-top image is white and there is no unidentified line word at the top of the fragment, let $a_1 = 0$; if not, continue to P2.

P2. If the first line word of the fragment is identified, similarly let $a_1 = 0$ (because the upper position of this line meets the upper boundary of the fragment); otherwise, if the first line word of the fragment is unidentified, let $a_1 = l_1$, where the l_1 is the lower position of the first unidentified line.

P3. If the border-bottom image is white and there is no unidentified line word at the bottom of the fragment, let $a_1 = 0$; if not, continue to P4.

P4. If the last line word of the fragment is identified, also let $a_4 = 0$ (because the lower position of this line meet the lower boundary of the fragment); else if the last line of the fragment is unidentified, let $a_4 = l_4$, where the l_4 is the upper position of the last unidentified raw.

P5. If there is any identified line word in the fragment, let $a_2 = l_2, a_3 = l_3$, where l_2 is the upper position of the last identified line at the bottom of the fragment and l_3 is the lower position of the last identified line at the bottom of the fragment. Meanwhile, let $l = l_2 - l_3$, where l is the word height. Let $l' = l_4 - l_3$, where l' is the height of inter-row space, then

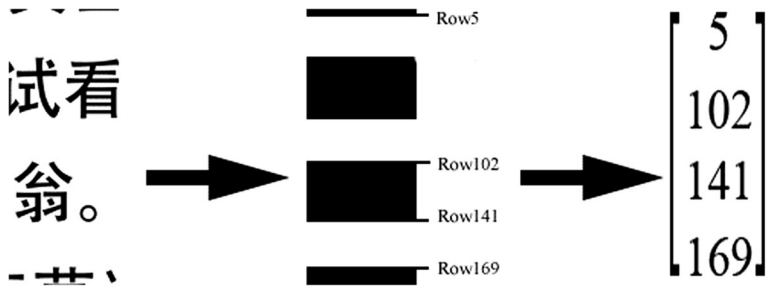


Fig. 3 The procedures of the feature extraction

continue to P6; if not, let $l_2 = 0, l_3 = 0$ with l, l' as the mean value from other fragments and continue to P7.

P6. If $a_3 < L - l - l'$, where L is the fragment height (because at the end of the paragraph, there may be no word at the end of line (as is visible in fragments 15 to 19 in Fig. 4 below), .so there is a need to modify the clustering vector), modify $a_2 = a_2 + l + l', a_3 = a_3 + l + l'$. Similarly, modify a_1 and a_4. End of procedure.

P7. If $a_1 = 0, a_4 = 0$, it means there is no text content in this fragment, so the fragment can be removed from the RCCSTD problem; if not, we need to modify a_2, a_3 according to a_1, a_4, l, l'.

P8. If a fragment was rotated 180 degrees, we execute procedure 1 though procegure 7 to get false clustering vector $CV' = \left[a_1', a_2', a_3', a_4'\right]^T$, the relationship between real clustering vector $CV = [a_1, a_2, a_3, a_4]^T$ and false clustering vector CV' is shown as follow: $a_1 = L - a_4'; a_4 = L - a_1'; a_2 = L - a_3' + l + l'; a_3 = L - a_2' + l + l'$, we can get the real clustering vector based on flase clustering vector.

After all of these procedures, it should be possible to obtain $\mathrm{CV} = [a_1, a_2, a_3, a_4]^T$.

2.1.2 The cluster center

The clustering vector of the first fragment in each row can be defined as the cluster center. The first fragment in each row has a number of notable features. For example, the left side of the fragment's image is white. This being the case, it is easy to discover the first fragment in each row and establish it's clustering vector. This clustering vector is then named as the cluster center. The cluster center of each row can be defined as $CV_1', CV_2' \ldots\ldots CV_m'$ (we assume that the text document is shredded into $m \times n$ fragments, so there are m cluster centers).

After establishing the cluster centers, the fragments need to be clustered into m classes. The criterion for this is $\left|CV_i - CV_j'\right| < T_{th}$, where T_{th} is a threshold. In other words, as long as the distance between the clustering vector CV_i and the clustering center CV_j' remains less than T_{th}, it means that the fragment i and the clustering center CV_j' are in the same row. If necessary, the few remaining fragments can be classified by hand.

2.2 Splice in row

Solving the RSSTD problem amounts to transforming a random arrangement of fragments into a correct arrangement of fragments. The reconstruction problem within rows can be modeled as a Traveling Salesman Problem (TSP). Fragments can be converted into vertexes in a graph and adjacent correlations of fragments can be converted into the edges of vertexes. The distance between two vertexes is large when the adjacent correlation of the two fragments is low. By the same token, the distance is small if the adjacent correlation is high. So solving the

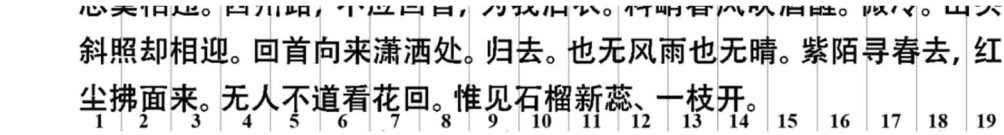


Fig. 4 A set of fragments that include ones where the clustering vector needs to be modified

RSSTD problem is equal to finding the shortest minimum Hamilton circle in a Complete Weighted Graph.

2.2.1 Setting up the TSP model

In order to obtain the adjacency matrix for the TSP it is necessary to calculate the distance (i.e. cost function) between any two vertexes. The distance between two fragments $d_l(i,j)$ can be defined as Eqs. (1)–(3).

$$d_l(i,j) = \sum_{y=1}^{L} e'(i,j,y) \tag{1}$$

$$e'(i,j,y) = \begin{cases} 1 & e(i,j,y) > \tau \\ 0 & otherwise \end{cases} \tag{2}$$

$$\begin{aligned} \mathrm{e}(i,j,y) = {} & 0.7\left|V_i(y)-V_j(y)\right| + 0.1\left|V_i(y+1)-V_j(y+1)\right| + 0.1\left|V_i(y-1)-V_j(y-1)\right| \\ & + 0.05\left|V_i(y+2)-V_j(y+2)\right| + 0.05\left|V_i(y-2)-V_j(y-2)\right| \end{aligned} \tag{3}$$

where $\mathrm{e}(i,j,y)$ represents the distance at point y between fragment i's right border and fragment j's left border. $e'(i,j,y)$ is the result of the binarization of $\mathrm{e}(i,j,y)$, and τ is a threshold that can be deduced through experience. Note that y in the equation needs to meet the following condition:

$$\mathrm{y} \in [3, L-2] \cap \mathrm{y} \in N^* \tag{4}$$

If it does not, another formula has to be used to calculate $e(i,j,y)$ as follow:

$$e(i,j,y) = \left|V_i(y)-V_j(y)\right| \tag{5}$$

After these calculations have been completed, the required adjacency matrix can be obtained and the RCCSTD problem for a row can be transformed into a more straightforward TSP. However, one further point needs to be noted: the definition of the distance is one of main sources of error in splicing because it is never possible to perfectly quantify the adjacent correlation between fragments.

2.2.2 The method for the TSP

The Traveling Salesman Problem is a classic problem in the field of combinatorial optimization. It has been proven to be a complete Non-Deterministic Polynomial problem. As a mature domain of research, there are now many possible solutions for a TSP. A number of papers suggest that the ant colony algorithm is a good solution to the TSP [12]. This algorithm is both highly repeatable and accurate, so this is the algorithm we have used to solve the TSP set out above.

2.3 Increase the splicing accuracy in row

As the number of fragments in a row increases, or the border text information for a fragment decreases, the splicing accuracy declines notably [9]. In order to improve the splicing accuracy, we propose the following two solutions: First of all, a better definition of the distance can be

found. A number of papers suggest ways in which this might be accomplished [5, 9, 26, 33]. In each of these pattern recognition is considered to be the best approach. Secondly, the TSP solution itself can be made more effective. As mentioned in the introduction to this paper, limited boundary information and a tremendous number of fragments can make it difficult to find an optimal solution for complex spatial configurations, and the results can be sensitive according to how the cost function is defined. In our view, increasing splicing accuracy by redefining the cost function is not the most efficient way to proceed. However, if it is possible to improve the effectiveness of the TSP solution by basing it upon the characteristics of the RCCSTD problem, this would appear to be a more promising way forward. We therefore focus upon the latter approach to increasing splicing accuracy in a row. There are two strategies that might be used to achieve this aim: a combination strategy, and a divide-and-conquer strategy.

2.3.1 The combination strategy

Some papers have pointed out (e.g. [6]) that when the number of cities declines in the travel Traveling Salesman Problem, the fault tolerance for distances in the TSP model increases. A related observation is that, as the number of fragments declines, the effect of the deviation between the definition of the distance and the adjacent correlation to splicing accuracy is weakened. Thus, reducing the number of fragments will improve the splicing accuracy for distances that are defined to be the same.

So how might the number of fragments be reduced? A specific set of operations that can accomplish this are as follows: Whilst the distance between fragments i and j is less than a specified threshold, the fragments i and j can be thought of as adjacent. The two fragments can then be merged into a new fragment (this process is illustrated in Fig. 5). The new fragment takes fragment j's left border as its own left border and takes fragment i's right border as its own right border. In this way the number of fragments is reduced, thereby improving the splicing accuracy.

2.3.2 The divide-and-conquer strategy

Usually blank spaces will appear at the ends of paragraphs. Thus, the quantity of information for fragments in the same row can be different. For example, in Fig. 3 fragments 1 to 14 have three lines of words, but fragments 15 to 19 only have two. If fragments within the same model have different quantities of information this can lead to mistakes and decrease the splicing accuracy.

The question arises in that case as to how to reduce the influence of this phenomenon. One of the best ways of tackling the problem is to think of a divide-and-conquer algorithm. By using this the fragments can be divided into 2 parts according to their information content. This not only improves the splicing accuracy, but also reduces the temporal complexity of the algorithm.

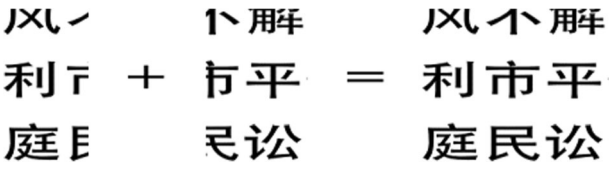


Fig. 5 A demonstration of fragment merging

2.4 Splice between row

When we have finished splicing in rows, the splicing between rows becomes an easy task. It can be achieved by matching the row fragments' clustering vectors (CVR_i). For the sake of clarity, the whole matching algorithm is shown below:

Algorithm. The row fragments matching algorithm.

Get the first and last row ID of the fragments named S_ID, E_ID;
C_ID= S_ID;(C_ID is the current fragment's ID for matching)
COUNT=1;(COUNT is the counter for the row fragments that have to be matched)
PATH=C_ID;(PATH is the splicing result)
while COUNT<=N **do**
I=1;**(** I is the current fragment's ID for being matched**)**
if the current fragment has no unidentified row at the bottom (a_4=0) **then**
    while $CVR_I(2) + CVR_{C_ID}(3) \neq$The interrow space height(l') **do**
        I=I+1;
    end while
else
    while $CVR_I(1) + CVR_{C_ID}(4) \neq$The word height($l$) **do**
        I=I+1;
    end while
end if
update the COUNT, PATH, C_ID;
end while

3 The experiments

The main point of this paper is to examine the viability and splicing accuracy of an algorithm that is capable of solving the RCCSTD problem. This being the case, we need to eliminate any interference arising from other factors so as to simplify the problem. There are many factors that can influence splicing accuracy beyond just the algorithm itself. These can include such things as; the loss of paper that is turned to dust by the shredding knives; the skew of the cuts relative to the text lines; the resolution of the scanner; noise in the image arising from the scanning process; and so on. Bearing this in mind, we created a test data set by using a digital simulation of a physical cross-cut shredder. The resolution of the text document's image was 1368 × 1980.

We considered a set of ten text documents. Five of them were in Chinese, and five in English. areal of the documents were in Times New Roman, 12 Font, with no line spacing. The documents were shredded into 5 × 5, 7 × 7, 8 × 8, 9 × 9, 10 × 10, 11 × 11, 11× 13,11 × 15,11 × 17 and 11 × 19 shreds by a computer to serve as our test data. So as to enable other research teams to replicate the experiment, we are using open data here to

demonstrate the process. This will be followed by the statistical results from the actual experiments, which made use of personal text documents.

We are going to use Appendix 3 of China's Undergraduate Mathematical Contest in Modelling (CUMCM)-2013, problem B [1], as our demonstration data. As space is limited, we will just use the most complicated parts of the document. The document was cut into 11 × 19 fragments. The size of the photo was 180 × 72 pixels, the size of a word in the photo was about 40 × 40 pixels, and the inter-row space height was about 28 pixels. We implemented our method in MATLAB R2012b and performed all tests on a Core i5 2450 M CPU with 4GB of RAM.

3.1 Demonstration of the clustering analysis and exception handling

The first step in the clustering analysis is to obtain the clustering vector, as described in the method presented above. However, as we begin the process, we find some mistakes in how the document was extracted, as shown in Fig. 6.

Due to the characteristics of Chinese, there are some interruptions in the vertical strokes of the word. This phenomenon exists in English too, but it is more obvious in Chinese. As is shown in Fig. 6, if there were no interruption in the extraction, the clustering vector we would obtain for this fragment would be $[7,119,159,0]^T$. However, the real clustering vector for the fragment is $[23,119,159,0]^T$. So we need to know how to modify this kind of error. These kinds of interruptions in vertical strokes can be thought of as a sort of one-dimensional Salt and Pepper Noise. This being the case, we can use an improved one-dimensional median filter to get rid of the noise. Based on the work presented in [34] and its associated experiment, the window length of the median filter is defined as 7, with the result after filtering being shown at the right of Fig. 6.

After dealing with the mistake, we obtain a clustering vector for each fragment and a cluster based upon the cluster center for each row. We set the clustering threshold, T_{th}, to be 1/20 of the vertical resolution of the fragments image that was used for the experiment. With this threshold the fragments can be clustered without mistake. The results are shown in Table 1.

3.2 Splicing result of demonstration data

As can be seen in the clustering results shown in the Table 1, we spliced in rows. The definition of the distance can then be used to obtain the adjacency matrix and the splicing problem can be transformed into a TSP. After this, the ant colony algorithm is used to solve the TSP. Referring to the literature [32], the parameters for the ant colony algorithm were set as $\alpha = 1$, $\beta = 5$, $\rho = 0.5$ and the number of ants was 19. The algorithm was then used to splice the

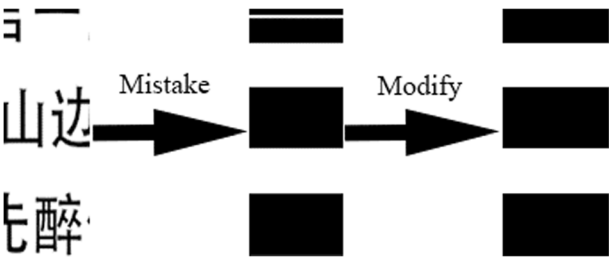


Fig. 6 Demonstration of clustering analysis and exception handling

Table 1 The clustering results

000	007	032	045	053	056	068	070	093	126	137	138	153	158	166	174	175	196	208
003	012	014	031	039	051	073	082	107	115	128	134	135	159	160	169	176	199	203
005	010	029	037	044	048	055	059	064	075	092	098	104	111	171	172	180	201	206
008	009	024	025	035	038	046	074	081	088	103	105	122	130	148	161	167	189	193
002	011	022	028	049	054	057	065	091	095	118	129	141	143	178	186	188	190	192
006	019	020	036	052	061	063	067	069	072	078	079	096	099	116	131	162	163	177
015	017	027	033	060	071	080	083	085	132	133	152	156	165	170	198	200	202	205
004	040	089	101	102	108	113	114	117	119	123	140	146	151	154	155	185	194	207
034	042	043	047	058	077	084	090	094	097	112	121	124	127	136	144	149	164	183
013	016	021	066	106	109	110	125	139	145	150	157	173	181	182	184	187	197	204
001	018	023	026	030	041	050	062	076	086	087	100	120	142	147	168	179	191	195

row fragments. In this way we were able to obtain a reconstruction of CUMCM-2013. problem B. Appendix 3 that was correct and that had been accomplished without manual intervention during the splicing (see Fig. 7). The algorithm took 16.43 s to reconstruct the document and an average of 1.43 s to splice each row.

3.3 Splicing result statistics of test data

Table 2 shows the results obtained by applying our method for the actual test documents. The number of fragments in a wrong position for each document in every shredded pattern is listed in the table. Note that the demonstration text document, labeled C1, is also included in the table.

It can be observed that there are two Chinese text documents that can be restored without any errors and without manual intervention across all patterns. At first sight it would appear that reconstructed Chinese text documents give a higher splicing accuracy than reconstructed English text documents.

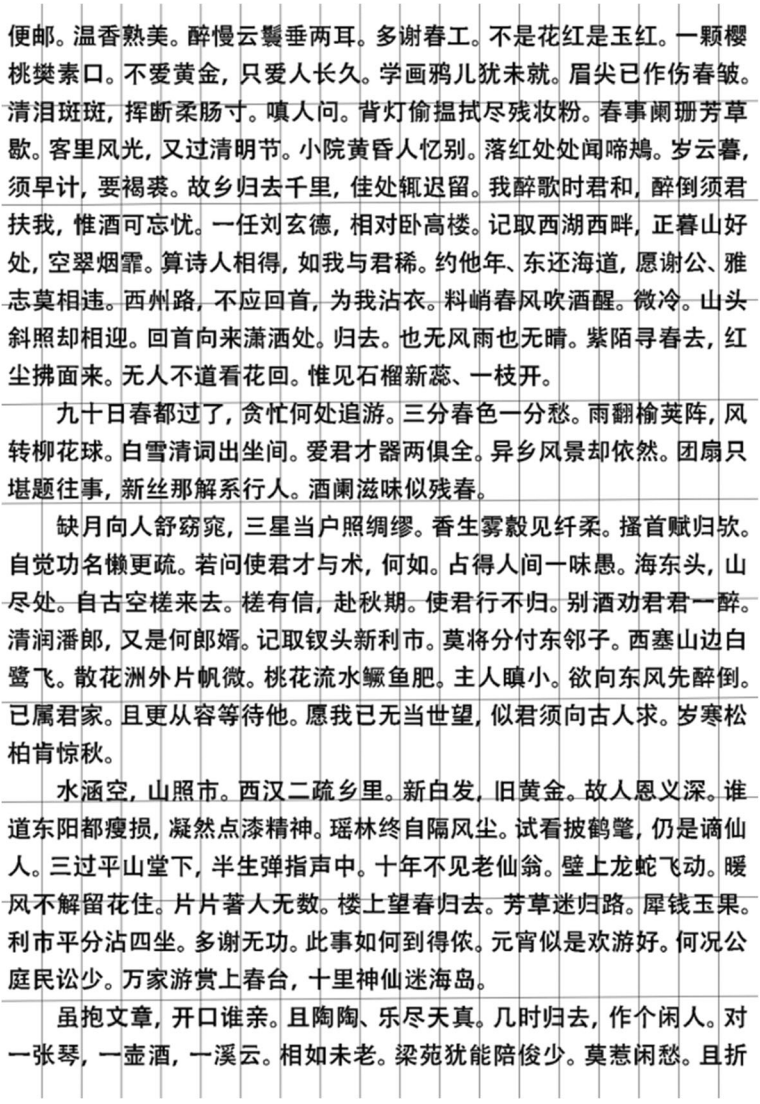
4 Discussion

4.1 The necessity of two strategies

One of the main concerns of this paper is to explore whether a combination strategy and a divide-and-conquer strategy can together improve the splicing accuracy in a row over the same defined distance. To examine this issue let us begin by looking at the difference between where these two strategies were or were not used in a series of tests. Here all of the test documents were shredded into 11 × 19 shreds, with document C1 continuing to be used for demonstration data.

4.1.1 Splice in row by basic model

First of all, we used the parameters for the ant colony algorithm presented in the section 3.2 as a basic model (without applying either of the potential strategies). The results we obtained for this basic model were as follow: there were 4 rows spliced correctly (Row 1, Row 3, Row 5 and Row 7) and 7 rows contained some mistakes (Row 2, Row 4, Row 6, Row 8, Row 9, Row 10 and Row 11). There were 28 mistakes overall within the 7 Rows that contained errors. For


便邮。温香熟美。醉慢云鬟垂两耳。多谢春工。不是花红是玉红。一颗樱桃樊素口。不爱黄金，只爱人长久。学画鸦儿犹未就。眉尖已作伤春皱。清泪斑斑，挥断柔肠寸。嗔人问。背灯偷揾拭尽残妆粉。春事阑珊芳草歇。客里风光，又过清明节。小院黄昏人忆别。落红处处闻啼鴂。岁云暮，须早计，要褐裘。故乡归去千里，佳处辄迟留。我醉歌时君和，醉倒须君扶我，惟酒可忘忧。一任刘玄德，相对卧高楼。记取西湖西畔，正暮山好处，空翠烟霏。算诗人相得，如我与君稀。约他年、东还海道，愿谢公、雅志莫相违。西州路，不应回首，为我沾衣。料峭春风吹酒醒。微冷。山头斜照却相迎。回首向来潇洒处。归去。也无风雨也无晴。紫陌寻春去，红尘拂面来。无人不道看花回。惟见石榴新蕊、一枝开。

九十日春都过了，贪忙何处追游。三分春色一分愁。雨翻榆荚阵，风转柳花球。白雪清词出坐间。爱君才器两俱全。异乡风景却依然。团扇只堪题往事，新丝那解系行人。酒阑滋味似残春。

缺月向人舒窈窕，三星当户照绸缪。香生雾縠见纤柔。搔首赋归欤。自觉功名懒更疏。若问使君才与术，何如。占得人间一味愚。海东头，山尽处。自古空槎来去。槎有信，赴秋期。使君行不归。别酒劝君君一醉。清润潘郎，又是何郎婿。记取钗头新利市。莫将分付东邻子。西塞山边白鹭飞。散花洲外片帆微。桃花流水鳜鱼肥。主人瞋小。欲向东风先醉倒。已属君家。且更从容等待他。愿我已无当世望，似君须向古人求。岁寒松柏肯惊秋。

水涵空，山照市。西汉二疏乡里。新白发，旧黄金。故人恩义深。谁道东阳都瘦损，凝然点漆精神。瑶林终自隔风尘。试看披鹤氅，仍是谪仙人。三过平山堂下，半生弹指声中。十年不见老仙翁。壁上龙蛇飞动。暖风不解留花住。片片著人无数。楼上望春归去。芳草迷归路。犀钱玉果。利市平分沾四坐。多谢无功。此事如何到得侬。元宵似是欢游好。何况公庭民讼少。万家游赏上春台，十里神仙迷海岛。

虽抱文章，开口谁亲。且陶陶、乐尽天真。几时归去，作个闲人。对一张琴，一壶酒，一溪云。相如未老。梁苑犹能陪俊少。莫惹闲愁。且折

Fig. 7 The reconstructed result for CUMCM-2013, problem B, Appendix 3

the purposes of discussion, we will focus on row 4 (see Fig. 8). The algorithm using the basic model took 3.23 s to splice a row on average.

4.1.2 Splicing in rows using the basic model and the divide-and-conquer strategy

As another test we used the basic model together with the divide-and-conquer strategy to splice the fragments. Once again Row 4 is used to illustrate the results (see Fig. 9). In this case the algorithm spent 1.98 s to splice a row on average.

As we can see, when the basic model is used together with the combination strategy, the number of mistakes declines significantly. So, the number of mistakes in Row 4 has decreased

Table 2 Splicing result statistics of test data

Patterns Documents	5 × 5	7 × 7	8 × 8	9 × 9	10 × 10	11 × 11	11 × 13	11 × 15	11 × 17	11 × 19
C1	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	3	4	4	7	11	20
C3	0	0	0	0	3	7	5	7	11	10
C4	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	2	6	9	6	12
Total of Chinese	0(5)*	0(5)	0(5)	0(5)	6(3)	13(2)	15(2)	23(2)	28(2)	42(2)
E1	0	0	0	2	5	9	11	17	20	20
E2	0	0	2	7	8	15	17	22	24	25
E3	0	0	0	5	8	13	18	21	23	27
E4	0	0	3	6	10	20	24	27	33	49
E5	0	0	0	0	5	8	14	20	25	27
Total of English	0(5)	0(5)	5(3)	21(1)	36(0)	69(0)	84(0)	107(0)	125(0)	148(0)

*0(5) means there are 0 fragments in a wrong position and 5 documents spliced correctly

from 6 to 2. In general, the total number of mistakes dropped from 28 to 23 and there were 2 more rows that were now spliced without errors (Row 8 and Row 10). However, this strategy is only useful when there are fragments in a row with different lines of text. In other word, this strategy cannot be used for every row.

4.1.3 Splice in row by basic model and combination strategy

We use the basic model and combination strategy to splice the fragments and use the Row 4 for demonstration (see Fig. 10). The algorithm with basic model and combination strategy will spend 1.80s to splice a row on average.

As we can see, when we are use the basic model and combine with combination strategy, the number of the mistakes is declining. For example, the number of mistakes in Row 4 decrease from 6 to 3. In generally, the total number of the mistakes is dropping from 28 to 7 and there are only 3 rows still have splicing mistake (they are Row 4, Row 8 and Row10).

4.1.4 Splicing in rows using the basic model, the combination strategy and the divide-and-conquer strategy

Finally, let us look at where the basic model, the combination strategy and the divide-and-conquer strategy where all used together to splice the fragments. Yet again we refer to Row 4 to demonstrate (see Fig. 11). This algorithm spent 1.43 s splicing a row on average.

As can be seen in Fig. 11, the number of mistakes in Row 4 drops to 0. The total number of mistakes also dropped to 0.

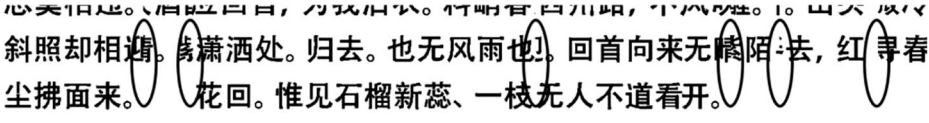


Fig. 8 Demonstration of the splicing results in the basic model using Row 4

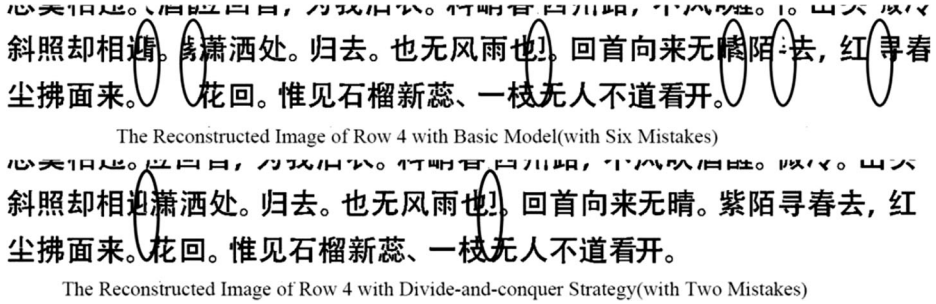


Fig. 9 Showing the difference between where the divide-and-conquer strategy is used or not used

4.1.5 Summarizes the overall differences between using or not using the two strategies

There is a table for contrasting the differences between use or not the two strategies (see Table 3). In order to exclude the possible influence of any accidental factors, we ran the same experiment across another 9 test documents. This produced some differences in the results between Chinese and English documents, so we list the results here separately (See Table 4).

Table 4 suggests that the efficiency of the two strategies is not the same for Chinese and English documents. Overall, they appear to be more effective for Chinese documents. However, based on the two tables together we can still conclude that the combination strategy and the divide-and-conquer strategy can not only improve splicing accuracy in a row but also reduce the temporal complexity for the algorithm. Thus, there is a clear and significant value in applying the combination strategy and the divide-and-conquer strategy to the RCCSTD problem.

4.2 Convergence of the algorithm

Convergence is an important property for algorithms, especially heuristic algorithms. An ant colony algorithm was used to solve the RSSTD problem, so it is necessary to test its convergence. This was done using numerical experiments. The parameters of the ant colony algorithm were set as $\alpha = 1$, $\beta = 5$, $\rho = 0.5$ and the number of ants was 19. The evolution curve for the optimal solution of the ant colony algorithm for Rows 4 and 5 in test document C1 is shown in Fig. 12.

As a result of the experiment detailed in Section 4.1 we know that the cost function for the optimal solution for Rows 4 and 5 is 360.7513 and 354.4565

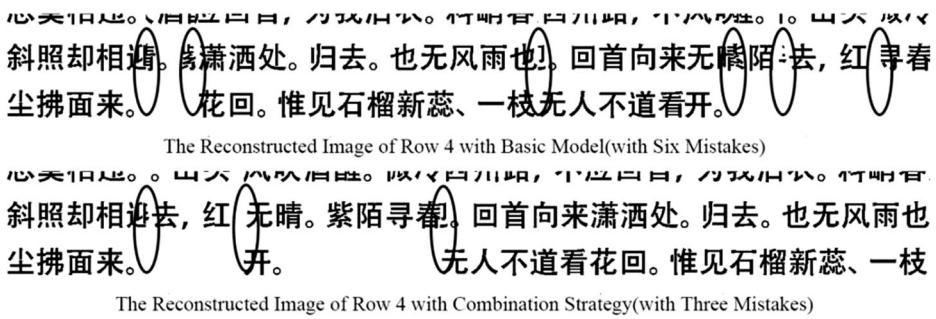
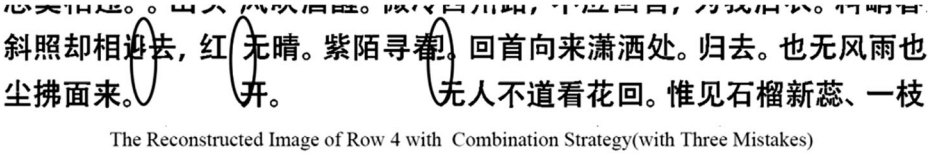
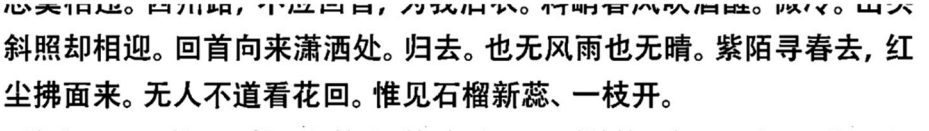


Fig. 10 Showing the difference between where the combination strategy is used or not used



The Reconstructed Image of Row 4 with Combination Strategy(with Three Mistakes)



The Reconstructed Image of Row 4 with Combination Strategy and Divide-and-conquer Strategy(Correct)

Fig. 11 Showing the difference between using just the combination strategy and using both strategies together

respectively. Looking at Fig. 12 it can be seen that our algorithm converges in the 25th and 31st iteration to get the optimal solution. Based on the discussion above, we would therefore argue that our algorithm is convergent.

4.3 Comparison between distance definitions

The distance definitions between the fragments could affect the splicing accuracy in a row. It is therefore necessary to test the differences between these distances. We compared our distance definition with the Manhanttan distance, the Euclidean distance and the cosine distance. The number of fragments in a wrong position and the number of correctly spliced documents are listed in Table 5 (all of the test documents were shredded into 11 × 19 shreds).

As we can see from Table 5, whilst our methods may not offer the best solution for splicing accuracy, on the basis of our test data our definition achieves the best results for correct document splicing (the effects of accidents cannot be ruled out). Furthermore, the proposed scheme is easy to calculate and understand, reinforcing our choice of this distance for the cost function in row splicing.

4.4 Comparing the splicing accuracy with Xu et al.'s paper [31]

This paper is derived in part from the work of Xu et al. It is therefore also necessary to compare our splicing accuracy with Xu et al.'s original results. In order to control the variables, we use the same definition of as Xu et al. put forward [31] which was as follows:

$$Accuracy = 1 - \frac{thenumberof fragmentsinwrongposition}{thetotalnumberof fragments} \tag{8}$$

Table 3 The differences between using or not using the two strategies

Compared items	Number of correct rows (Row ID)	Number of mistakes	Run time on average (splicing one row)
Different models			
Basic model	4(1,3,5,7)	28	3.23 s
Basic model + divide	6(1,3,5,7,8,10)	23	2.98 s
Basic model + combination	8(1,2,3,5,6,7,9,11)	7	1.80s
Basic model + combination + divide	11(1,2,3,4,5,6,7,8,9,10,11)	0	1.43 s

Table 4 Experimental results using another 9 test documents

Diffident model	Basic model	Basic model + divide	Basic model + combination	Basic model + divide + combination
Experiment results				
Number of mistakes for Chinese documents	127	111	63	42
Number of mistakes for English documents	239	224	170	148
Run time on average	3.22 s	2.92 s	1.89 s	1.47 s

We calculated the splicing accuracy for both Chinese and English text documents. The results are shown in Fig. 13.

We found that the splicing accuracy was more than 0.95 for the Chinese documents and 0.85 for the English ones. This means that reconstructing Chinese text documents gives a higher splicing accuracy than it does for English text documents. There are many reasons for this phenomenon. One is that Chinese text document fragments have more boundary content than English text document fragments when they are both using the same font. Chinese also has a strong "square" quality to its characters, especially in the modern simplified form of Mandarin. Accuracy rates for English documents are in the mid-80% range while the number of fragments raised to 150, it is may not a good result for classic document recognition tasks.

Setting aside the difference between the two languages, note that the accuracy was higher than 0.85 for either case, when the number of fragments was lower than 210. Outside of this, note that the scope of line k1 is −0.0016 bigger than the scope of line *k*2 (−0.000318) in absolute value. Line *k*1 represents the splicing accuracy of English text documents that were shredded into fragments of 8 × 8, 9 × 9, 10 × 10, and 11 × 11. Line k2 represents the splicing accuracy of English text documents that were shredded into fragments of 11 × 13, 11 × 15, 11 × 17, and 11 × 19. This means that the splicing accuracy is more likely to be affected by the

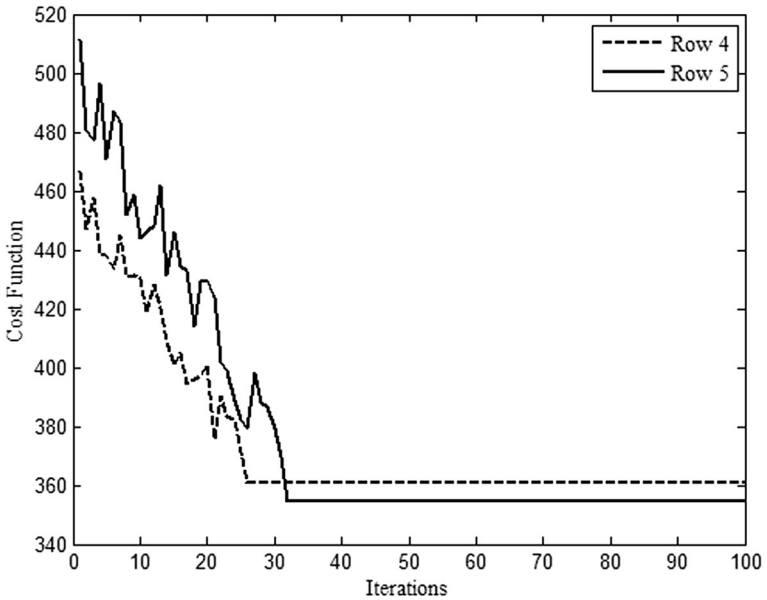


Fig. 12 Evolution curve for the optimal solution for our ant colony algorithm

Table 5 Comparative results for distances

Distances	Manhanttan distance	Euclidean distance	Cosine distance	Our definition
Documents				
Chinese documents	41(1)**	37(1)	57(0)	42(2)
English documents	142(0)	178(0)	133(0)	148(0)

**41(1) means there are 41 fragments in a wrong position and 1 document spliced correctly when a Manhanttan distance is used in the model

number of rows than it is by the number of fragments in a row. In other words, the splicing accuracy for our approach decreases more quickly as the number of rows in a shredded text document increases, rather than the number of fragments in a row.

Comparing our results with Xu et al.'s (see Fig. 14), the picture shows that our proposed algorithm gives a higher splicing accuracy than Xu et al.'s approach, especially when the number of fragments is greater than 100.

5 Conclusion and future work

The algorithm presented in this paper was able to achieve a set reconstruction task - the automatic reconstruction of document CUMCM-2013, problem B, Appendix 3 - without error. A first point of note is that using a combination strategy and a divide-and-conquer strategy together can improve splicing accuracy whilst reducing temporal complexity. A second point of note is that the approach presented in this paper gives a higher splicing accuracy than the approach proposed in similar previous work by Xu et al. [27] because it is not sensitive to the number of fragments in a row. The splicing accuracy was over 0.95 for Chinese test data and over 0.85 for English test data. Thus the splicing accuracy is generally high. However, our approach does have some issues: 1) the test data was not physically cross-cut shredded text documents and this may make a difference, and we overlook many factors which can affect splicing accuracy such as abnormal printing on paper, inclined cutting of paper, dust by the blades and so on; 2) the splicing accuracy may not be sensitive to the number of fragments, but it *is* sensitive to the number of rows; 3) the narrow margins of the documents will help us identify the first and last fragments in each row, this characteristic make our task easier, not all of the shredded documents have narrow margins; 4) if there are two or more identical cluster

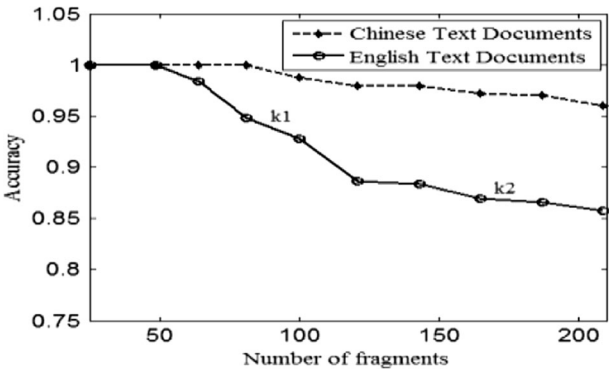


Fig. 13 Splicing accuracy when reconstructing English and Chinese documents shredded into different pieces

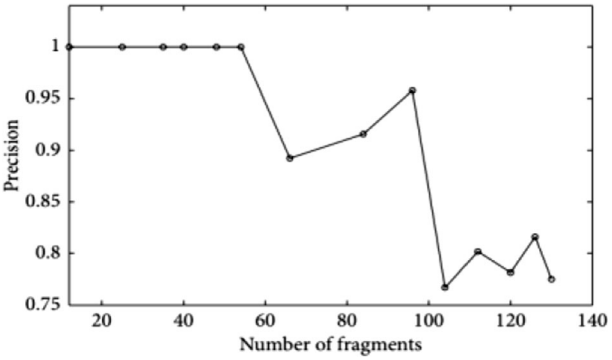


Fig. 14 The precision of reconstructing a document shredded into different pieces according to the work of Xu et al. [31]

centers in one shredded document, it is not possible to finish the clustering and the fragments cannot be reconstructed. The latter issue suggests that the algorithm may not be effective for the reconstruction of multiple shredded documents because of the possibility of identical cluster centers.

In future work we will be looking at how to extract more features to improve the accuracy of the row clustering and will be attempting to address the various issues described above, especially pay more attention on reconstruction cross-cut shredded multiple text document based on simulate data and real data. On the other hand, we need to focus more on reconstruction real shredded text document, there are many issues need to be solve during transforming the real data image to the simulate image which was used in our paper. To sum up, above and beyond all other matters, maintaining a high splicing accuracy is the primary goal when tackling the RCCSTD problem, and this has been the central focus for our approach.

Acknowledgements This research was supported by the National Science Foundation of China (11172016, 11472022, 11772016). The authors would like to thank the reviewers of this paper for their constructive and thoughtful comments. The authors thank Editsprings (www.editsprings.com) for its linguistic assistance.

References

1. China Undergraduate Mathematical Contest in Modelling (2013) CUMCM-2013 contest problems [WWW document]. URL http://en.mcm.edu.cn/problem/2013/2013_en.html. Accesses on 31 July 2017
2. Cho TS, Avidan S, Freeman WT (2010) A probabilistic image jigsaw puzzle solver[C]. Computer Vision and Pattern Recognition. IEEE, pp 183–190
3. Cui J, Liu Y, Xu Y et al (2013) Tracking generic human motion via fusion of low- and high-dimensional approaches. IEEE Trans Syst Man Cybern Syst 43(4):996–1002
4. Goldberg D, Malon C, Bern M (2002) A global approach to automatic solution of jigsaw puzzles. Comput Geom Theory Appl 28(2):165–174
5. Gong YJ, Ge YF, Li JJ et al (2016) A splicing-driven memetic algorithm for reconstructing cross-cut shredded text documents. Appl Soft Comput 45:163–172
6. Huang HS (2005) Study on new methods to solve traveling salesman problem. Tianjin University, Tianjin (Chinese)
7. Justino E, Oliveira LS, Freitas C (2006) Reconstructing shredded documents through feature matching. Forensic Sci Int 160(2):140–147

8. Kosiba DA, Devaux PM, Balasubramanian S et al (2002) An automatic jigsaw puzzle solver[C]. Iapr International Conference on Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing, vol 1. IEEE, pp 616–618
9. Lin HY, Fan-Chiang WC (2012) Reconstruction of shredded document based on image feature matching. Expert Syst Appl 39(3):3324–3332
10. Liu Y, Zhang X, Cui J et al (2010) Visual analysis of child-adult interactive behaviors in video sequences[C]. International Conference on Virtual Systems and Multimedia. IEEE, pp 26–33
11. Liu Y, Cui J, Zhao H et al (2012) Fusion of low-and high-dimensional approaches by trackers sampling for generic human motion tracking[C]. International Conference on Pattern Recognition. IEEE, pp 898–901
12. Liu Y, Nie L, Han L et al (2015) Action2Activity: recognizing complex activities from sensor data[C]. International Conference on Artificial Intelligence, pp 1617–1623
13. Liu Y, Nie L, Liu L et al (2016) From action to activity: sensor-based activity recognition. Neurocomputing 181:108–115
14. Liu L, Cheng L, Liu Y et al (2016) Recognizing complex activities by a probabilistic interval-based model[C]. Thirtieth AAAI Conference on Artificial Intelligence, pp 1266–1272
15. Liu Y, Zheng Y, Liang Y et al (2016) Urban water quality prediction based on multi-task multi-view learning[C]. 25th International Joint Conference on Artificial Intelligence, pp 2576–2582
16. Liu Y, Zhang LM, Nie LQ, et al (2016) Fortune teller: predicting your career path[C]. Thirtieth AAAI Conference on Artificial Intelligence, pp 201–207
17. Lu Y, Wei Y, Liu L et al (2016) Towards unsupervised physical activity recognition using smartphone accelerometers. Multimed Tools Appl:1–19
18. Ma L (2002) Reviews on the algorithm of traveling salesman problem. Mathematics in Practice and Theory 30(2):156–165 (Chinese)
19. Ma J, Zhao J, Tian J et al (2014) Robust point matching via vector field consensus. IEEE Trans Image Process 23(4):1706–1721
20. Ma J, Qiu W, Zhao J et al (2015) Robust L2E, estimation of transformation for non-rigid registration. IEEE Trans Signal Process 63(5):1115–1129
21. Pomeranz D, Shemesh M, Benshahar O (2011) A fully automated greedy square jigsaw puzzle solver[C]. Computer Vision and Pattern Recognition. IEEE, pp 9–16
22. Prandtstetter M (2009) Hybrid optimization methods for warehouse logistics and the reconstruction of destroyed paper documents [D]. Vienna University of Technology
23. Prandtstetter M (2009) Meta-heuristics for reconstructing cross cut shredded text documents[C]. Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July. DBLP, pp 349–356
24. Prandtstetter M, Raidl GR (2008) Combining forces to reconstruct strip shredded text documents[M]. Hybrid Metaheuristics. Springer, Berlin
25. Preotiuc-Pietro D, Liu Y, Hopkins DJ et al (2017) Beyond binary labels: political ideology prediction of twitter users[C]. The 55th annual meeting of the Association for Computational Linguistics, pp 1–12
26. Schauer C, Prandtstetter M (2010) A memetic algorithm for reconstructing cross-cut shredded text documents[C]. International Conference on Hybrid Metaheuristics. Springer-Verlag, pp 103–117
27. Sleit A (2013) An alternative clustering approach for reconstructing cross cut shredded text documents. Telecommun Syst 52(3):1491–1501
28. Ukovich A, Ramponi G, Doulaverakis H et al (2004) Shredded document reconstruction using MPEG-7 standard descriptors[C]. IEEE International Symposium on Signal Processing and Information Technology. IEEE, pp 334–337
29. Wang Y, Ji DC (2014) A two-stage approach for reconstruction of cross-cut shredded text documents[C]. Tenth International Conference on Computational Intelligence and Security. IEEE Computer Society, pp 12–16
30. Wolfson H, Schonberg E, Kalvin A et al (1988) Solving jigsaw puzzles by computer. Ann Oper Res 12(1): 51–64
31. Xu HD, Zheng J, Zhuang ZW, Fan S (2014) A solution to reconstruct cross-cut shredded text documents based on character recognition and genetic algorithm. Abstr Appl Anal:1–12
32. Yan YN (2008) Parameter optimization of ant colony algorithm and its application. Nanjing University of Science and Technology, Nanjing (Chinese)
33. Zhao B, Zhou Y, Zhang Z et al (2014) Information quantity based automatic reconstruction of shredded Chinese documents[C]. IEEE, International Conference on TOOLS with Artificial Intelligence. IEEE Computer Society, pp 1016–1020
34. Zhou J (2007) Improved algorithm of median filter in image processing [D]. Beijing University of Posts and Telecommunications, Beijing (Chinese)

Junhua Chen is a master degree candidate in Beijing University of Technology, College of Life Science and Bioengineering, His major research direction is image processing.

Youjun Liu is a professor in Beijing University of Technology, College of Life Science and Bioengineering, His major research direction is image processing, biomechanics.