A study of automatically assembling paper fragments on simulated annealing algorithm

Xian Liang[1], Haiyun Qin[1] , Jianpeng Shi[1], Lisa Huang[2], and Jiatai Gang[3, *]

[1] College of Mechanical Engineering, Dalian University, Dalian 116622, China

[2] Portacom New Zealand Limited, Auckland, 1642, New Zealand

[3] College of Information Engineering, Dalian University, Dalian 116622, China

*Corresponding author: GANG Jiatai, Email:gjt1960@126.com

Keywords: Grey value matrix, Character features, Edge goodness of fit, Simulated annealing algorithm.

Abstract. In this paper, a study is conducted on assembling paper fragments with regular edge geometry, using the example of rectangular pieces. During the assembly process, visual information of the fragments' is digitally extracted by MATLAB software to achieve the corresponding grey value matrix; edge characteristics are derived and used to find fragments which belong in the first column. Adjacent rows are matched subsequently according to similarity and an annealing algorithm is used to join the paper fragments by setting edge goodness of fit as the objective function. From practice, it is found that the above method increases assembly efficiency as well as accuracy. This is of significant value to the relevant fields of fragment assembly.

Introduction

Automatic fragment assembly technology is an important application used broadly in various fields of file recovery, including judicial evidence recovery, historical documentation and malfunction analysis. Recently, through restoration of the Stasi files, the study of automatic fragment assembly has perked wide public interest[1].

There are large amounts of existing research on irregular fragment assembly technology, for example Leitao & Stolfi's use of incremental dynamic programming sequence matching[2]. Luo[3] suggested using an edge detection algorithm based on linking scanned line segments to compose a closed curve of fragment edges; line and circle interpolation methods are adopted to guarantee scanning direction continuity. He[4] used intelligent algorithms to match fragments by contour similarities, by ant colony optimization. Zhao[5] highlighted deficiencies of classic assembly methods based on geometry features, and achieved effective semi-auto assembly by text analysis.

The above research all target fragments of irregular geometries. With the emergence of paper shredders, fragments have steadily become more regular[6]. In comparison, irregular fragments can be assembled by matching contour features whereas regular fragments are more difficult to assemble due smooth edges. Currently, few research on assembling regular paper fragments exist. Zhao[7] digitally extracted pixels from vertically cut, regular fragments to derive into corresponding matrix. Match rate is then set as objective function to assemble fragments by method of exhaustion.

This paper re-assembles rectangular paper fragments cut vertically and horizontally, by digitally extracting the fragments' visual information with MATLAB software to achieve the corresponding grey value matrix. Contour characteristics are derived and used to find fragments which belong in the first column. Adjacent rows are matched subsequently according to similarity and an annealing algorithm is used to join the paper fragments by setting edge goodness of fit as the objective function.

Digitalizing fragment visual information.

As the paper fragment contours are regular, contour features cannot be used in the assembling process. Digital information of the fragments visual features are extracted and analysed to achieve reassembly.

The smallest resolution cell of an image is a pixel[8], each image has $m \times n$ cells, m representing length and n width. The corresponding grey value matrix A has m rows and n columns, (m, n) representing the pixel's position within the image, equivalent to (x, y) in a co-ordinate system. $(m, n) = x_{m,n}$, represents the grey value of (m, n) within the range of 0~255 (0 is black, 255 white, the rest a transition grey scale). Each fragment is scanned as an image and imported to MATLAB to construct a $m \times n$ grey value matrix as shown in Fig. 1:

$$A = \begin{bmatrix} 0 & 0 & 39 & \cdots & \cdots & 255 & 47 & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 255 & 48 & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 255 & 48 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & \cdots & \cdots & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 255 & 255 & 255 & \cdots & \cdots & 255 & 255 & 255 \\ 255 & 255 & 255 & \cdots & \cdots & 255 & 255 & 255 \\ 255 & 255 & 255 & \cdots & \cdots & 255 & 255 & 255 \end{bmatrix}_{m \times n}$$

Fig. 1 A fragment image and its corresponding grey value matrix

Thus, every pixel can be translated into a numerical matrix which constitutes the image. When an image is in grey scale, the imread function in MATLAB can be used to convert pixel information into a corresponding matrix. Using MATLAB to generate the grey value set S to digitize the paper fragment's visual features, when the number of fragments equal $M \times N$, set S is $M \cdot m \times N \cdot n$, matrix A represents a single paper fragment within set S.

Assembling algorithm.

The criterion of fragment assembly and restoration is the goodness of fit between edge pixels of adjacent fragments; a better fit means a higher probability of adjacency. The calculation process is based on that of TSP[9], turning the problem of paper fragment assembly into that of combinational optimization. The edge pixel goodness of fit is set as objective function, and through simulated annealing algorithm, auto assembly of paper fragments is achieved. The process is represented as in Fig. 2:

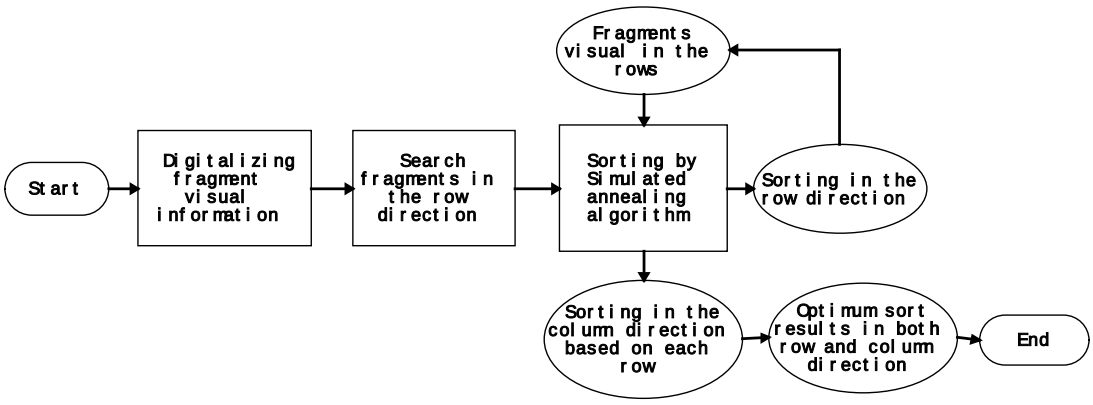

Fig. 2 Auto assembly process of paper fragments

A sketch of the Algorithm. Simulated annealing algorithm[10] is a random optimization algorithm based on the Monte Carlo method; it is a combination of the metallurgical annealing process and combinational optimization. Through random search within preset parameters, the algorithm is able to probabilistically avoid local optimum solutions and approach the global optimum.

Process of the Simulated annealing algorithm is as shown in Fig. 3:

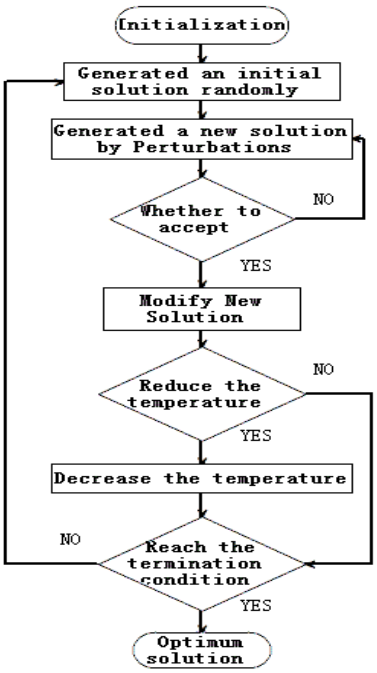

Fig. 3 process of Simulated annealing algorithm

Determining edge fragments of the original document and fragments by row. In this study, the original document is separated into 11 rows and 19 columns, thus $11\times19=209$ rectangular fragments as shown in Fig. 4. Each fragment has a grey value matric of 180×72 . Generic documents have blank margins with a grey value of 255. By searching for white edges, edge fragments of the original document can be first located.

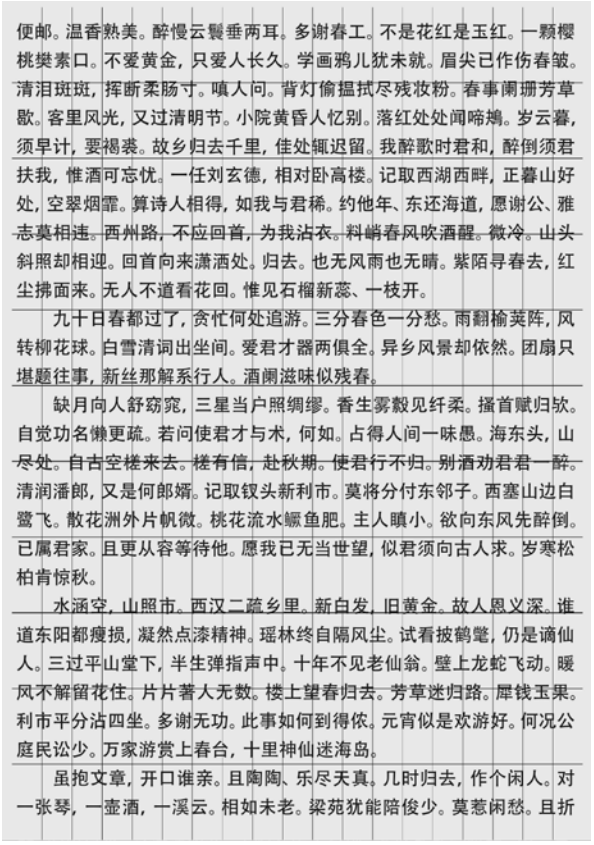

Fig. 4 A document page cut into 209 regular fragments

With example of searching for 255 grey values on the left edge, for every image with a pixel matrix of 180×72, the search range is pinpointed to the 180×4 matrix element on the far left, if the matrix has sum that is a multiple of 255, the fragment has a blank left edge. Through this method, it was found that there are exactly 11 fragments each for the far left column 1 and far right column 19.

It was discovered that there were 22 and 35 fragments which satisfied the criteria for having a blank top edge and blank bottom edge respectively, contradicting the reality of 11 fragments on each edge. This means the above search method does not yet suffice, and further searches are required according to text line spacing. 1 fragment from 11 of those in column 1 is analyzed with line spacing and its relative vertical position within the fragment as bounding condition, filtering fragments in the same row. As shown in Fig. 5, three fragments have three identical line spacing, and identical positions within the fragment. By searching for similar information, the 19 fragments of that row can be located. The same method can be applied to locate the remaining 10 rows.

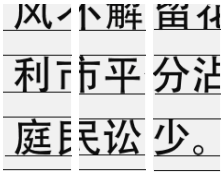

Fig. 5 Three fragments in the same row

Search process for fragment rows. According to the above analysis, the implemented method of identifying each fragment's serial number within its row is as follows:

Step 1: Selecting a fragment randomly within the first 11, and find fragments of the same row by using its line spacing information.

Step 2: Find the sum of pixel point values of each row within an image to get a 180×1 column matrix.

Step 3: Determine if the 180 numbers of the column matrix can be divided completely by 255, if so record as 1, if not, as 0, thus establishing a 0-1 column vector sized 180×1.

Step 4: Find sum of the remaining 208 pixel point values and determine whether each value can be divided completely by 255, establishing a 0-1 column vector.

Step 5: From the far left 0-1 column vector, randomly select 2 ~ 5 sections of consecutive 1 values, as the defining region for the row's line spacing. Other fragments of the same row are found through defining those images with the same defining regions in the same positions.

Step 6: Randomly select another fragment from the first column, repeat steps 2~5 until 19 fragment serial numbers are found for each row.

Fragment assembly by row. The four corner fragments can be found through searching edge fragment serials. When searching for the upper left corner fragment, the top and left edge are both blank, its serial number should be the intersection elements of the left most column and top row. Similarly, the other 3 corners can be found.

Serial numbers of the four corner fragments exist both within their respective column sets and row sets, therefore can be identified from the edge fragment serials.

Arrange the 19 paper fragments of each row using a simulated annealing algorithm. For two adjacent fragments calculate the logical value of pixel matrix $S_{i,j}(k,72)==S_{i,j+1}(k,1)$, ($k=1,2,\cdots,180; i=1,2,\cdots,11; j=1,2,\cdots,18.$), setting as the match rate of these fragments. When all fragments are assembled, larger the sum of match rates, the more accurate is the assembly. Fig. 6 and Fig. 7 are partial edge information of adjacent fragments within a row.

Fig. 6 Two fragments in the same row

255	255
255	255
255	200
246	39
73	0
0	0
0	0
0	3
0	156
149	255
255	255
255	255
255	255
255	255

Fig. 7 Marginal grey value matrix of adjacent fragments within a row

Implemented method of identifying the serial number of each fragment within its row through simulated annealing algorithm is as follows:

Step 1: From the above search results, extract serial numbers of a fragment row. The first and last fragments are already identified.

Step 2: Use the Monte Carlo iteration to generate an initial feasible solution index $x(0)$ as the initial solution.

Step 3: Initiate the programme from the current temperature T_i.

Step 4: Set match rate of the right most edge pixel value on fragment j and left most edge pixel value on fragment $j+1$ as F, and set the objective function to maximize the global match rate. Calculate the objective function value F_0.

$$M\text{ax}\ \ F=\sum_{j=1}^{18}\left(S_{i,j}(k,72)==S_{i,j+1}(k,1)\right) \tag{1}$$

Of which, $k=1,2,\cdots,180; i=1,2,\cdots,11; j=1,2,\cdots,18$.

Step 5: Use the initial solution as core to create stochastic disturbance within the solution space. Generate a perturbation solution x', and calculate its objective function F'.

Step 6: Determine if the obtained perturbation solution satisfies formula (2). If the generated function value F' of x' is greater than F_0 of $x(0)$, then accept $x(1)=x'$ as a new solution of this iteration; otherwise, accept x' as the new solution with a probability of $e^{\frac{f(x')-f(x(0))}{T}}$.

$$G(F_0 \to F') = \begin{cases} 1 & f(F') < f(F_0) \\ e^{\frac{f(F')-f(F_0)}{T_0}} & \text{Other} \end{cases} \tag{2}$$

Of which, $x(k), k=0,1,2...n$ is the initial solution, x' the perturbation solution.

Step 7: After running multiple iterations in temperature T_i, cool by mechanism

$$T' = \alpha \times T \tag{3}$$

Of which, $\alpha \in (0,1)$, stipulating α=0.999, T is the current temperature, T' is the temperature of the next cooling phase, $T_0 = 1$ is the initial temperature.

Step 8: Determine if the termination temperature $e = 10^{-20}$ is satisfied, if $T \le e$, terminate the cooling mechanism and algorithm. If not, repeat steps 3~7.

By optimizing the arrangement of 19 fragments in each row, the near-optimum solution for each row can be found. Fig. 8 is the assembled image of a fragment row, thus is the implemented process of assembling paper fragments by row using simulated annealing algorithm.

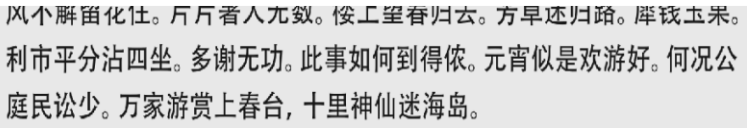

Fig. 8 auto assembing result of a row

Fragment assembly by column. The assembly of paper fragments by column is similar to that by row, the process also implemented through simulated annealing algorithm. As when assembling by column there are more edge pixels, the process is more accurate and more efficient.

When the row and column assembly is complete, restoration of the original document can be achieved. Though the assembly process in this study used document with line spacing as example, the process can also be used for documents without.

Conclusion

In this paper, a study is conducted on assembling paper fragments with regular edge geometry. Rectangular fragments are scanned as images, the visual information digitally processed into corresponding grey value matrix and used to analyze the fragment characteristics. Line spacing and spacing position is set as constraints to find matching fragments within the same row; the edge pixel match rate are set as objective function to find the optimum solution through simulated annealing algorithm. Fragments are assembled within each row, and then vertically row by row.

The study not only considers the match rate of edge pixels but also visual characteristic to amplify assembly accuracy. Simulated annealing programme is used to transform the challenge of automatic assembly into that of set optimization, increasing assembly efficiency through macro management. However it is notable that in comminuted cases, the method used cannot guarantee high accuracy due to lack of visual information on each fragment, and in turn will affect assembly efficiency.

References

[1] X. Zhang, Y.L. Bu, L.J. Zhu, Zongtan Zhou. Computer Simulation,2006,23(11), in Chinese.

[2] Da Gama Leitao, H.C.,Stolfi, J. IEEE Transactions on Pattern Analysis and Machine Intelligence,2002,24(9).

[3] Z.Z. Luo. Chinese Journal of Scientific Instrument ,2011,32(2), in Chinese.

[4] P.F. He, Z.T. Zhou, D.W. Hu. Computer Engineering & Science,2011,33(7), in Chinese.

[5] Z.Z. Luo. Computer Engineering and Applications,2012,48(5), in Chinese.

[6] K.J. Zhao. Digital Technology and Application,2010(6),in Chinese.

[7] J. Zhao, P.P. Li. Computer CD Software and Applicatio,2013,(20), in Chinese.

[8] E.W. Wu, D.M. Zhou, D.F. Zhao. Journal of Yunnan University,2007,29(5), in Chinese.

[9] H.D. Zhu, Y. Zhong. Computer technology and development, 2009, 19(6): 32-35, in Chinese.

[10] Reinelt G. ORSA journal on computing, 1991, 3(4): 376-384.