Image-Based Techniques for Shredded Document Reconstruction

Huei-Yung Lin and Wen-Cheng Fan-Chiang

Department of Electrical Engineering,
National Chung Cheng University,
168 University Rd., Min-Hsiung
Chia-Yi 621, Taiwan, R.O.C
lin@ee.ccu.edu.tw, hdtonestep@yahoo.com.tw

Abstract. This paper proposes an image-based technique for shredded document reconstruction. The problem is different from solving jigsaw puzzles since curved boundaries and color information are not available. Currently most research on document recovery focuses on image feature exaction and analysis. In this work, we present a complete procedure which is capable of reconstructing a full page of shredded document. Similarity measure based on shred boundary correlation is defined for pattern matching. A weighted digraph is then used to derive the final shred sorting result. Experiments are presented for both the synthetic and real datasets.

1 Introduction

One of the essential problems in digital image processing is the reconstruction of damaged images. In the past few decades, a large number of computational algorithms have been proposed to deal with restoration of degraded images [1,2]. The sources of degradation are commonly modeled by image acquisition noise (e.g., optical defocus and atmospheric turbulence blur), or data transmission noise (e.g., interference between different channels) [3]. In either case, the spatial relationship between pixels in an image is assumed to be available, and most of the existing techniques are focused on the recovery of the photometric aspect of the original image.

There are, however, other classes of image defects which are caused by splitting an image into several pieces. Jigsaw puzzle can be thought as one common example of this type of *damaged* images. The recovery process is usually to assemble the small pieces of a fragmented image based on their contour shapes or contents, such as texture or color information [4,5]. For more general cases, the objective of fragmented image recovery is to find the best subimage arrangement which resembles the original image. Thus, the underlying reconstruction issues are no longer part of the classic image restoration problem, but belong to an object recognition and classification problem. Moreover, the solution to this problem usually involves pattern matching and graph theory.

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 155–166, 2009.

This paper aims to address the problem of shredded document recovery using image-based techniques. It is not only an interesting research topic, but also has many applications on forensics and investigation science [6]. Although sometimes considered as a special case of jigsaw puzzle [7], this problem actually preserves different characteristics and requires its own solving strategy. In the past few decades, many researchers focused on developing optimal solutions to the jigsaw puzzle problem, but fairly little work has been done for shredded document analysis. Recently, due to the huge demand for document reconstruction, this issue has attracted the attention of government agents and private companies for extensive investigation [8,9,10,11]. However, to the authors' best knowledge, there are still no standard techniques or complete system description available in the literature.

In this work, we present the computational algorithms for shredded document recovery. The boundaries of the shredded document are assumed to be straight and indistinguishable, and only the interiors are used to verify the correctness of the assembled fragments. Moreover, the texture information on the shred boundaries might be lost due to the shredding noise. In our two-stage approach, image-based techniques are first used to evaluate the similarity between any pair of shreds, followed by a graph-based algorithm to derive the best shred sorting result in terms of a locally shortest path. The proposed method using the shred coding scheme and average word length is insensitive to the shredding noise on the image boundaries. Experimental results are presented for both the computer generated and real scanned shredded documents.

2 Image-Based Similarity Evaluation

The proposed shredded document reconstruction approach consists of the following five stages: image acquisition and pre-processing, special shred selection, shred coding, similarity measure, and graph-based sorting.

2.1 Image Acquisition and Pre-processing

Shred images for reconstruction are acquired by scanning the shredded document placed on a blue background, followed by object segmentation and length normalization in the shredding direction. Although some texture details might be lost during the normalization process, the computational complexity for pattern matching in the subsequent stages is greatly reduced. To remove the saw-tooth shape noise on the boundaries caused by the paper shredder and the shading caused by scanning, a one-dimensional morphological erosion is carried out in the horizontal direction (i.e. orthogonal to the shredding direction). Finally, the resulting shred images are binarized and the image features are extracted for document reconstruction.

One of the important prerequisites for correct pattern matching between the shred images is to align the text lines across all available pieces. This text and non-text region separation is achieved by segmenting the histogram obtained

from the horizontal projection of each shred image. Furthermore, the local maxima of the horizontal projection histogram are used to identify the top-lines and base-lines of the text lines [12]. These features will be used later to identify the relationship between the shreds in the shred coding stage.

2.2 Special Shred Selection

For a general shredded document there usually exist three types of special pieces, which are different from the majority of the shreds. They are namely the blank (or all-white) shreds, and the leftmost and rightmost shreds containing the text part of the original document.

The blank shreds commonly appear near the borders or on the separation of a multiple column document. Since there is no text information available by definition, they can be freely removed from the document reconstruction process. The leftmost shred is characterized by the one containing no texture near its left border but with texture near or on its right border. Vice versa for the definition of the rightmost shred. It is clear that these two types of shreds can be easily verified by examining the histogram of vertical projections (i.e. along the shredding direction). Thus, they are singled out first and served as the starting and ending vertices in the following graph-based shred sorting stages.

2.3 Shred Coding

From the histogram of horizontal projections, each shred image consists of a number of text blocks separated by several disconnected blank blocks. If we compare this binary pattern with the one generated from the original document image, it can be seen that the text blocks of any individual shred is a subset of those in the original document. Furthermore, there might be different text block patterns for different shred images mainly due to the large space introduced by the beginning or ending of a text line. For the shreds with high spatial proximity, however, those patterns can be identical or only differ by a few text blocks.

Based on the above observation, a shred coding scheme is proposed to group the closely related shreds. The idea is to assign similar binary coded patterns to the shreds based on their spatial proximity. This grouping method can significantly reduce the computational complexity, especially for document reconstruction from a large number of shred images. The algorithm consists of first creating a shred model from all of the shred images, followed by binary coding for the individual shreds.

Since the shred model contains all possible text block locations of the individual shred images, it can be constructed by taking the union of the horizontal projections of all shred images. Let the projection distribution of shred i be $p_i(j)$ for $i = 1, 2, \ldots, n$, where j is a variable along the shredding direction, then the shred model is represented by the set

$$M = \{j \mid \sum_{i=1}^{n} p_i(j) > th, 1 \leq j \leq m\} \tag{1}$$

where th is a threshold and m is the length of the shred images in pixel.

Due to acquisition noise, quantization error, or slight miss alignment between the shreds, the projection histogram might not provide perfectly separable text blocks. Thus, the base-lines of the text regions are further used to robustly indicate the locations of the text blocks. The k-th text block of the shred model is then given by

$$B(k) = \{j \mid j \sim b_k, 1 \leq j \leq m\} \tag{2}$$

where $\sim$ represents the connectivity relation and b_k is the k-th base-line from the top.

The binary encoding for each shred is accomplished by comparing its text block or base-line locations with the shred model. Since the text block pattern of an individual shred is merely a subset of the shred model, a "0" or "1" will be assigned depending on whether the text block of a shred is absent or present on the model. More specifically, the k-th bit of an n-bit binary code c can be written as

$$c_k = \begin{cases} 1, & \text{if } \exists\ j \text{ such that } j \sim b_k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where the number of bits n is equal to the number of text block in the shred model.

2.4 Similarity Measure

In shredded document reconstruction, a similarity measure is a metric to evaluate the similarity between any two shreds. Higher score on the similarity measure generally means higher correlation between the pair of shreds. Based on this, a probability distribution from the shred permutation can be derived and used to recover the correct shred order in the original document.

In this work, we propose two approaches for the similarity measure computation. One is to use the discrepancy in the shred coding result, and the other is to calculate the correlation between the shreds based on the average word length. It should be noted that, for the shred images from a single-sided document with correct orientation (i.e. all shreds with top-down or bottom-up text), two similarity computations should be carried out between any two shreds since there are two effective boundaries for each of them.[1]

Shred Coding Discrepancy

The shred coding pattern described in the previous section can be though of as a simplified representation of the document layout. Based on the continuity characteristic of the document content, a negative correlation is assigned to each bit difference between the binary codes. Consequently, there is a negative

[1] If the shred images are not oriented, then there will be four and eight similarity computations between any pair of shreds for a single-sided and a double-sided sheet of document, respectively. Moreover, the computational complexity is increased exponentially for multiple-sheet documents. Both cases are not discussed in the current work.

correlation score between any pair of shreds, which serves as one of the similarity measures for sorting shreds to the correct order in the original document.

Different from both-side aligned documents, the space between two words in a text line is constant for general left-aligned or right-aligned documents. As a result, the binary codes for these classes of documents have the property that the shreds with the same code are very likely to belong to the same group in general cases. Furthermore, the smaller bit difference between the binary codes means that the corresponding shreds might be spatially closer to each other. In other words, the shred coding result plays a major role on a coarse level similarity check.

It is clear that the shreds with the same binary code form a unified pattern group, and no further discrepancy exists due to the highest correlation score (i.e. zero) between each other. Thus, a second sorting scheme is required exclusively for each group of the same binary coded shreds. Since this stage is a refinement of the coarse level similarity check, there is usually a limited number of shreds in each group for the similarity measure computation.

Average Word Length

The second similarity measure proposed in this work is based on the average word length of a general document. Under the assumption that the length of each word in a document should be as close to the average word length as possible, a negative correlation score can be evaluated using the difference. Although the word lengths are not constant in a document, this similarity measure is valid for a general probability distribution of word length, especially with a large sample size.

For each shred permutation in the same binary coded group, the negative correlation score based on the average word length is defined as the summation of the difference between a word length and the average word length. More

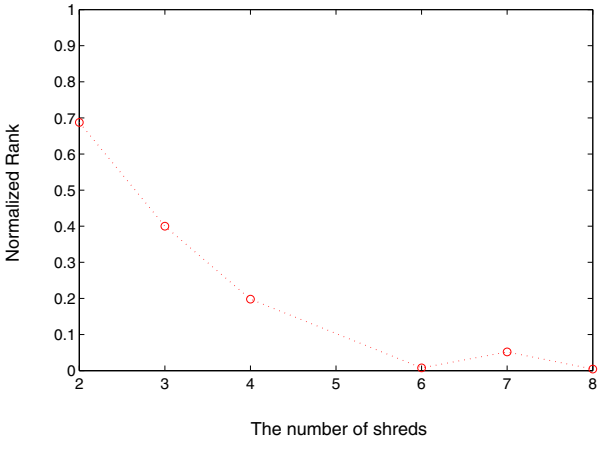

Fig. 1. The (normalized) rank of the correct permutation versus the number of shreds in a group for a simulation result. The correct permutations are of relatively high ranks for the groups with a large number of shreds. The result is given by the average of ten computer generated documents.

specifically, suppose the document contains m text lines (which can be identified by the shred model described previously) and there are n shreds in a group, then the similarity measure for a specific shred permutation is given by

$$-\sum_{p=1}^{m}\sum_{q=1}^{\alpha_p}|w_{p,q}-\bar{w}| \tag{4}$$

where $\bar{w}$ is the average word length, $w_{p,q}$ and α_p the q-th word length and the number of words in the p-th text line, respectively. The objective is to find the shred permutation, say indexed as j, from the $n!$ possible permutations that maximizes the similarity measure, i.e.

$$j = \arg\max_i g(i) \tag{5}$$

where $g(i)$ is the correlation score of the i-th shred permutation defined by Eq. (4), and $i = 1, 2, \ldots, n!$. The shred permutation given by Eq. (5) is then used to recover the shred order in the binary coded group.

Ideally, the correlation function $g(i)$ is maximized by the correct permutation of the shreds under the assumption of constant word length. For a general document with variable word lengths, however, high correlation score only implies that the shred permutation result is more reasonable. As an example of the same binary coded group from computer generated documents, Fig. 1 illustrates the statistics of normalized ranks of the correlation scores associated with the correct permutation for various numbers of shreds. Although the correct permutations do not possess the top rank using Eqs. (4) and (5), they are still of relatively high ranks for the groups with a large number of shreds. Thus, the figure indicates that the proposed average word length approach is feasible, especially when the number of shreds increases. By assigning a suitable threshold on the normalized rank, it is guaranteed to cover the correct permutation.

As suggested by the simulation result given in Fig. 1, Table 1 lists the reasonable thresholds versus the number of shreds in a group adopted in the implementation. Note that the threshold is assigned as the rank among all shred permutations instead of the correlation score. It might also be concluded from the table that the rank of $(n-1)!$ is a conservative choice if the number of shreds n in a group is small. This is a good rule of thumb since the shred coding in

Table 1. The thresholds versus the number of shreds adopted in the implementation, where n is the number of shreds in a group. The maximum number of n is given by the number of shreds in the document. In this case, only a single binary coded pattern is provided by the shred coding stage.

Number of shreds	Rank of reasonable threshold
$2 \sim 3$	$n!/2$
4	$n!/3$
5	$n!/5$
above 6	$n!/10$

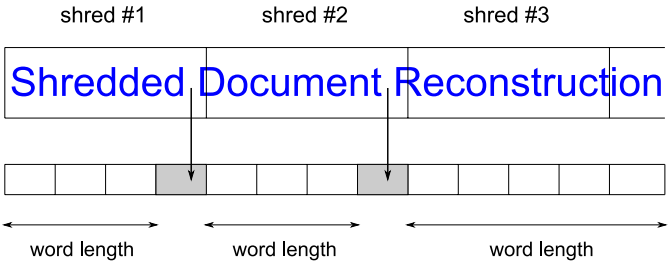

Fig. 2. A shred is partitioned to 4 strips and the strip width is used as the unit of word length. Note that the space between two words always occupies one unit strip width.

the coarse level similarity check usually results in a small number of shreds in a group (less than 10 in general).

Since the character size might not be the same for different documents or even varies in a document, it is not suitable to use pixel as the unit of word length. To make the word length distribution less dependent on the metric unit, each shred is further partitioned to several strips and the strip width is set as the unit for correlation score calculation. An example is illustrated in Figure 2, where each shred is further partitioned to four strips. Note that the space between two words always occupies at least one unit strip width, and the word length is rounded to an integer strip width. This quantization process can greatly reduce the computation cost while maintain the correctness of threshold settings.

3 Graph-Based Sorting Scheme

The objective of shredded document reconstruction is to sort the unorganized shred images and recover the correct order in the original document. Based on the grouping result from shred coding and the similarity measure, document reconstruction problem is modeled as a weighted digraph (directed graph). More specifically, the shred images are represented by the vertices of the graph, and the correlation scores between the pairs of shred images are assigned as the weighted edges of the graph.

Since each shred image has two boundaries (left and right) when merged with another shred image, directed edges for both the left-right and right-left adjacency relations are assigned to each vertex. Finding the shortest path connecting the starting and ending vertices is then equivalent to selecting the optimal shred permutation (with the fixed leftmost and rightmost ones) for document reconstruction.

First Stage Sorting

Without any prior knowledge of the shred characteristics, the shredded document reconstruction problem should be modeled as a complete graph since the similarity between any pair of shreds has to be evaluated. The required computation therefore grows exponentially as the number of shreds increases. Because

the improper pairings based on the similarity measure are usually inevitable, the correctness of the reconstruction results will also degrade due to the larger number of inaccurate similarity evaluations.

In this work, a two-stage sorting scheme is proposed to reduced the high computational complexity and mis-pairing rate introduced by a large size complete graph. In the first stage sorting, a simplified digraph is created based on the shred coding result. Each vertex in the graph is modeled as a supernode representing the set of the same binary code. The weighting on the directed edges is defined by the number of bit difference between the pair of binary codes. Since the starting and ending vertices are available from the special shred selection, the shortest path can be easily determined sequentially by the set of minimal weighted edges.

Although rarely happened in practice, there might be a tie on the bit difference between two pairs of binary codes. In this case, the continuity of the bit pattern is further used to determine the best match. Let s be the number of bit pattern change defined as the number of transitions from 0 to 1 or 1 to 0 in a shred image. Suppose A is the set of shred images which have the same number of bit difference when connected to shred i for pairing, then the best match is given by

$$\arg\max_{j}(s_j - s_i) \tag{6}$$

where $j \in A$. If the ambiguity still cannot be resolved, then the method described in the next stage will be applied on this coarse level sorting.

Second Stage Sorting

The second stage sorting focuses on finding the shortest path of the digraph associated with the supernode derived from shred coding. Except for the supernodes containing the border shreds (i.e. the leftmost and the rightmost), the starting and ending vertices in the same binary coded set are not available. One simple way to obtain the shortest path is to compute the cost function or the similarity metric exclusively for all possible permutations of the shred images. The computational cost of this brute-force approach is obviously too expensive for a large number of shreds.

In this work, the "shortest" path is generated sequentially by identifying the two adjacent vertices connected by the directed edge with the highest weight in the same coded group. Although the link between any two vertices is bipartite, merging the adjacent vertices using the highest weighted edge will simultaneously removes the possibility of path finding using the other edge. Continue this process of merging the adjacent vertices, the edges for the shortest path is identified and the digraph is shrunk to a single vertex corresponding to the supernode of the coded group. This approach does not guarantee the true shortest path as given by, for example, the Hungarian method used for the assignment model [13]. However, the proposed algorithm is easy to implement and provide the sub-optimal results in most cases.

Note that finding a path using this approach might not result in an ordered set of directed edges during the path creating process. However, the required

sorting for the shred images is independent of the edge selection or location orders. More specifically, let w_{ij} represents the weight from vertex i to vertex j where $i \neq j$. Note that w_{ij} is not equal to w_{ji} in general. Then the first edge is given by connecting vertices p_1 and q_1, where

$$(p_1, q_1) = \arg \max_{i,j,i \neq j} w_{ij} \tag{7}$$

and the r-th edge is given by connecting vertices p_r and q_r, where

$$(p_r, q_r) = \arg \max_{i,j,i \neq j} \{w_{ij} | i \neq p_1, \ldots, p_{r-1}, j \neq q_1, \ldots, q_{r-1}\} \tag{8}$$

The set of edges (p_r, q_r) for $r = 1, \ldots, n$, where n is the number of shreds in the same coded group, forms a sub-optimal short path.

In the implementation, an $n \times n$ correlation matrix associated with the bipartite graph is created based on the relationship between any pair of shreds in an n-shred group. This matrix is not symmetric in general, because there are two possible permutations and therefore two different correlation scores for each pair of shreds. The proposed method can be implemented efficiently as follows:

i) Find the maximum weight, w_{ij}, in the matrix. The corresponding directed edge (i, j) is added to the path.
ii) Cross out all entries belonging to the i-th row and j-th column in the matrix.
iii) Go to Step i) and repeat until w_{ij} is the last entry in the matrix.

The above algorithm automatically set the starting and ending vertices as those connected by the least weighted edge, i.e. the last entry remaining in the matrix.

Three-dimensional ego-motion estimation has been one of the most important problems for the application of computer vision in mobile robots.

Accurate estimation of ego-motion is very helpful for human computer interaction and short-term control such as braking, steering, and navigation.

In the past, there have been many methods which use flow vectors as the basis of their derivations for motion estimation.

No matter their derivations are linear or nonlinear, the flow vectors are observed by using single camera.

However, there are some drawbacks on using only one camera.

First, one can only solve the translation up to the direction, i.e., the absolute scale cannot be determined.

This is the well known scaling factor problem.

Second, the size of view field substantially affects the accuracy

Fig. 3. The document used for reconstruction (Rotated 90° to fit in the page).

4 Experiments

A computer generated document image as shown in Fig. 3 is used for the experiments. The unit strip width is set as 1/3 of the average shred width, which is used for the similarity measure based on the average word length. To distinguish two consecutive words by quantized word length as shown in Fig. 2, the word spacing is set as 6 pixels. The average word length is estimated prior to the reconstruction, with 5 and 6 units for the synthetic and real images, respectively.

For the shredded document reconstruction from scanned images, the original document is printed out to an A4 paper and then shredded to 21 pieces (excluding the blank ones) with each 7 mm wide. They are then scanned and normalized

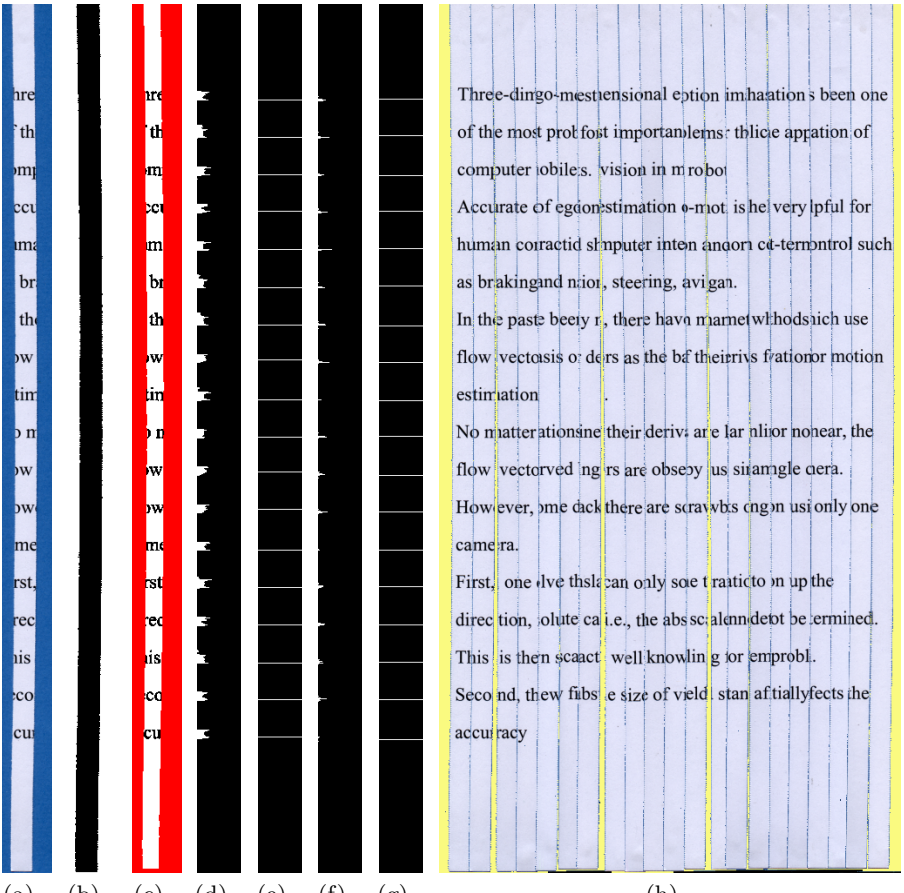

Fig. 4. The intermediate image pre-processing results of one scanned shred. The same procedure from (a) to (e) is carried out for all shred images. The histogram of text base-lines (f) from all shreds is then used to create the shred model (g). (h) shows the shredded document reconstruction result using scanned shred images.

to an image with 1000 pixels high. Similar to the synthetic dataset, the shred images are indexed by their original order: $0, 1, 2, \ldots, 20$.

In the image pre-processing stage of the real shreds, the procedure described in Section 2.1 is carried out for feature extraction. Fig. 4 illustrates the intermediate image pre-processing results of one scanned shred. The original shred scan, foreground segmentation, the image after morphological erosion, the histogram of horizontal projections, and the base-line locations of the text lines are shown in Figs. 4(a) – 4(e), respectively.

In the special shred selection, the leftmost border shred is properly identified as shred 0 for this left-aligned document. To generate the shred model for shred coding, the base-line histogram as shown in Fig. 4(f) is obtained by summing the base-line image of each shred. Fig. 4(g) shows the base-lines of the shred model created by taking the local maxima of the base-line projection histogram. Based on the shred coding results, the correct grouping, $\{0, 1, 2\} \rightarrow \{3\} \rightarrow \{4, 5, 6, 7, 8, 9, 10, 11, 12\} \rightarrow \{13\} \rightarrow \{14, 15, 16\} \rightarrow \{17, 18\} \rightarrow \{19\} \rightarrow \{20\}$, is obtained using the first stage sorting.

The similarity measure used for the second stage sorting is calculated with the following settings. Each shred is partitioned to 3 strips, the average word length is set as 6 units in terms of strip width, and the word spacing is set as 6 pixels. The threshold setting for a given number of shreds in a group is based on Table 1. The vertex merging algorithm described in Section 3 is carried out for the second stage sorting, and the final permutation is derived as $0-1-2-3-8-9-12-4-5-6-7-10-11-13-14-16-15-17-18-19-20$. Fig. 4(h) shows the reconstruction result. The number of discontinuities in this experiment is 8, out of the maximum of 20 possibilities.

5 Conclusion

In this work, we have presented an image-based technique for shredded document reconstruction. Several features of shred images are extracted for reconstruction with two similarity measures. The proposed algorithm using the shred coding scheme and average word length is insensitive to the shredding noise on image boundaries. A weighted digraph is then carried out to derive the optimal shred sorting result for document reconstruction in terms of the shortest path. Experiments are presented for both the synthetic and real data sets. The results show that the proposed method have correctly merged the majority of the shredded document.

Acknowledgment

The support of this work in part by the National Science Council of Taiwan, R.O.C, under Grant NSC-96-2221-E-194-016-MY2 is gratefully acknowledged.

References

1. Banham, M., Katsaggelos, A.: Digital image restoration. IEEE Signal Processing Magazine 14(2), 24–41 (1997)
2. Loce, R., Dougherty, E.: Enhancement and Restoration of Digital Documents: Statistical Design of Nonlinear Algorithms. In: Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, WA, USA (1997)
3. Gonzalez, R., Woods, R.: Digital Image Processing, 2nd edn. Prentice-Hall, Englewood Cliffs (2001)
4. da Gama Leitao, H., Stolfi, J.: A multiscale method for the reassembly of two-dimensional fragmented objects. IEEE Trans. Pattern Analysis and Machine Intelligence 24(9), 1239–1251 (2002)
5. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. Comput. Geom. 28(2-3), 165–174 (2004)
6. Justino, E., Oliveira, L.S., Freitas, C.: Reconstructing shredded documents through feature matching. Forensic Science International 160(2-3), 140–147 (2006)
7. Zhu, L., Zhou, Z., Hu, D.: Globally consistent reconstruction of ripped-up documents. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(1), 1–13 (2008)
8. Brassil, J.: Tracing the source of a shredded document. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 387–399. Springer, Heidelberg (2003)
9. Smet, P.D., Bock, J.D., Philips, W.: Semiautomatic reconstruction of strip-shredded documents. In: Said, A., Apostolopoulos, J.G. (eds.) Image and Video Communications and Processing 2005, vol. 5685, pp. 239–248. SPIE (2005)
10. Ukovich, A., Ramponi, G.: Features for the reconstruction of shredded notebook paper. In: International Conference on Image Processing, pp. III: 93–III: 96 (2005)
11. Biswas, A., Bhowmick, P., Bhattacharya, B.: Reconstruction of torn documents using contour maps. In: International Conference on Image Processing, III: 517–III: 520 (2005)
12. Lu, S., Chen, B., Ko, C.: Perspective rectification of document images using fuzzy set and morphological operations. Image and Vision Computing 23(5), 541–553 (2005)
13. Kuhn, H.: The Hungarian method for the assignment problem. Naval Research Logistics 52(1), 7–21 (2005)