Strip Shredded Document Reconstruction using Optical Character Recognition

Johannes Perl, Markus Diem, Florian Kleber and Robert Sablatnig

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology
Austria

Keywords: Document Reconstruction, OCR, Local Features

Abstract

Document reconstruction affects different areas such as archeology, philology and forensics. A reconstruction of fragmented writing materials allows to retrieve and to analyze the lost content. Due to the complexity of reconstruction, automated algorithms are necessary. A reconstruction methodology for shredded documents is presented in this paper which recognizes characters at the stripes' borders and matches them subsequently. In order to achieve this, an Optical Character Recognition (OCR) system is exploited, that is capable of recognizing partially visible characters by means of local features. Thus, no binarization needs to be performed. Preliminary results show the ability of matching shredded documents using the information of cut characters.

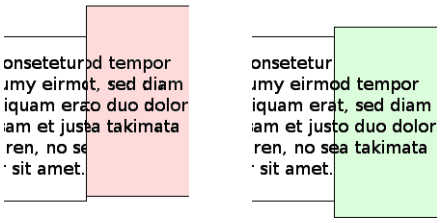
1 Introduction

Document reconstruction is the reassembly of single fragments of a document which have been destroyed either on purpose or accidentally. Documents can be torn by hand, shredded by a paper shredder, or can be destroyed by e.g. fire or water. Intended destruction of text documents is usually due to criminal motives or the protection of sensitive data.

Recent approaches focus on fully automated solutions as document reconstruction becomes intractable for humans for a large amount of fragments (complexity see [1]). Another possibility is to use a semiautomatic solution to allow the user to correct failures of the algorithm [2]. In this paper we will not consider any preprocessing step, but mainly focus on the reassembly of synthetic fragments using Optical Character Recognition (OCR), to show the systems capability of classifying and matching cut characters.

Schauer et al. [3] define three important subdomains within document reconstruction: (1) strip shredded documents, (2) manually torn documents and (3) cross-cut shredded documents. Although these subdomains can be considered to be a variation from typical jigsaw puzzles, they significantly differ in detail [3]. Cross-cut shredded documents represent a special case of strip shredded documents, as both share the characteristics that all resulting fragments possess the same size and boundary. Due to the missing significant difference in the fragments' contours, approaches concentrate on the visual content of the pieces [3, 4, 5]. In contrast to this the reconstruction of torn documents utilizes the characteristics of the abitrary shapes to find adjacent fragments [6, 7]. Contour matching of manually torn paper fragments comprises problems like gaps or overlaps, which can appear due to shearing effects [8].

The aim of this paper is to show whether OCR is able to improve document reconstruction by extracting the semantics of the visual content in the border region of fragments. By using OCR an extended border region is taken into account, looking for cut off characters to match on two fragments' contours. As visual features in the border region are considered, the approach presented in this paper is suitable to process shredded document fragments. Figure 1 shows an incorrect and a correct match of 2 synthetic stripes. It can be seen that the matching can be solved if the semantic information is taken into account.



(a) incorrect match (b) correct match

Figure 1. Two matching possibilities

This paper is organized as follows: Section 2 reviews the state-of-the-art in document reconstruction. In Section 3 the methodology of the proposed system is discussed, while Section 4 presents the results of the OCR applied to synthetically generated images. Finally Section 5 concludes the paper.

2 Related Work

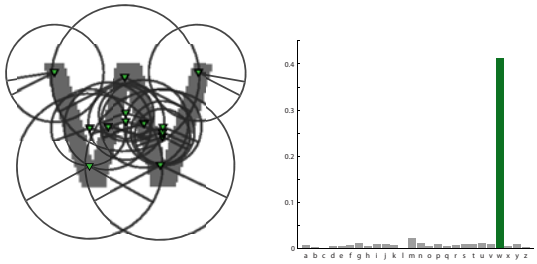
This section provides a short overview on methods trying to solve the reassembly of torn or shredded fragments.

Stieber et al. [6] are using contour matching to be able to reconstruct ruptured documents. They keep short contour seg-

ments, which have been shown to contain sufficient matching information by Leitão et al. [9], which are then used for the matching procedure. Another approach that reconstructs manually torn pages by using chain codes and Minkowski sums can be found in Biswas et al. [7]. In contrary to the former two approaches Ukovich et al. have focused on reconstructing strip shredded documents. One approach of Ukovich et al. [5] is using MPEG-7 descriptors, whereas in [4] they calculate 12 low level features (e.g. line spacing, document layout) from the visual information of each strip. The reconstruction approach used by Schauer et al. [3] focuses on the visual information in the border region ignoring essential information beside the border.

3 Methodology

Diem and Sablatnig [10] introduced an Optical Character Recognition (OCR) for ancient manuscripts that is capable of recognizing partially visible characters. The character recognition system is based on Scale Invariant Feature Transform (SIFT) features. Thus, for each character multiple SIFT features are computed which are then classified by multiple Support Vector Machines (SVMs). Each SVM decides if the currently observed character part is of a specific character class or not. Hence, a probability histogram consisting of 26 bins is constructed, where each bin indicates the likelihood of the current character part belonging to the respective class. In order to recognize the whole character, the probability histograms of the local measurements are accumulated to a stronger "global" probability histogram. This methodology is suitable for the reconstruction of shredded documents since the training is carried out on automatically detected parts of characters such as junctions, stroke endings or circles which allows for the recognition of cut characters. In Figure 2(a) the calculated SIFT features for the character 'w' and in Figure 2(b) the voting output for the character can be seen.



(a) (b)

Figure 2. (a) SIFT features of 'w' (b) voting output by OCR

Within this paper, the performance of the OCR method on cut images of characters, and hence a reconstruction using the voting output (see Figure 3) is analyzed.

3.1 Classification of Character Pieces

To evaluate the capability of the OCR to reconstruct single characters, they were cut horizontally and vertically by their center. Figure 4 shows examples of such pieces.

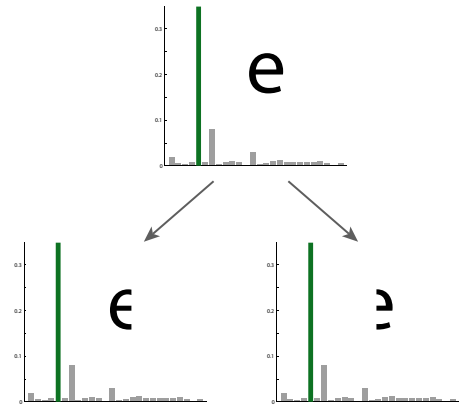
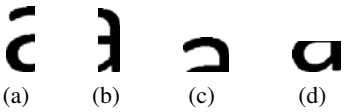


Figure 3. Matching of cut characters using the voting of the OCR approach



(a) (b) (c) (d)

Figure 4. Cut character a (a)(b) vertical cut (c)(d) horizontal cut

The two pieces of each cut character are then classified by the OCR. On the synthetic images the clustering step is skipped, since on each image only one known character is present. The results of the single character evaluation can be found in Section 4.1.

3.2 Reconstruction of Strip Shredded Documents

With the approach described in the previous section single characters were cut by their center. To simulate a strip shredder, synthetic tests on cut paragraphs have been performed. The paragraphs are cut through an arbitrary vertical plane.

The evaluation of the OCR algorithm with these strip shredded fragments is performed on the one hand against all other fragments and only within the same page on the other hand. This was done to obtain the overall performance, as well as the performance for a (in this case perfect) subset, which is achieved by clustering similar fragments prior to matching. Furthermore the tests were executed with strips containing different numbers of lines (5, 10, 15, 20, 25, 30 or 35).

Fragment processing In the first step the page strips are reduced to the characters touching the cutting edge. For the isolated character image the SIFT-features are calculated, the interest points are clustered together by the lines and the respective class probabilities are computed. Figure 5 shows the original paper stripes as well as the isolated character images and the corresponding line probability histograms.

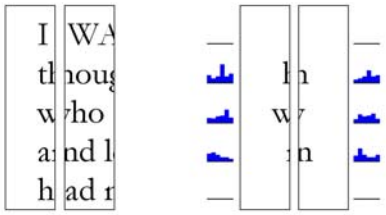


Figure 5. Two fragments before and after fragment processing

Fragment matching Having extracted the probability histograms, the perfect match for each fragment is searched for by calculating the Euclidean distance between the probability histograms for each line. Lines without interest points and therefore without probability information are refused and are not taken into consideration. For any stripe a different number of lines might be obtained, producing a bigger distance when a larger number of matches is found. Hence the Euclidean distance obtained is adjusted by calculating either the mean or the median. In addition tests were executed by normalizing the probability histograms into the range of [0 1]. This scales the most probable class to the value 1 and votes against matches having a different maximal class, improving the matching performance for characters which are cut near their center.

Multiple matches onto one fragment are not possible, thus if a fragment is already matched, the next best unmatched fragment is used. This procedure is called "Exclusive Matching" below. Exclusive Matching causes the order of processing the fragments to play an important role. Thus random optimization is used to find an "optimal" solution. After having calculated the distances between the fragments, the stripes are chosen by random and matched onto the best available candidate. This procedure is repeated multiple times to increase the probability to get a solution with the smallest global distance between all fragments.

4 Results

The subsequent experiments will show limitations and capabilities of the proposed system in the context of document reconstruction.

For the first experiment single character images were cut horizontally and vertically at their center respectively and the resulting pieces were classified by the OCR. The evaluation discussed in Section 4.1 targets at showing the general ability of the proposed system to recognize cut characters.

In Section 4.2 the system is evaluated by reconstructing synthetically shredded pages. The resulting paper strips are matched onto each other and a global optimal solution is searched for, using the extracted OCR features. Additionally the system's ability to find the ideal matching position of two differently sized paper strips is tested.

4.1 Classification of Character Pieces

Prior to the classification the OCR method is trained with all characters of the alphabet and 20 different fonts. With the SVM classifiers trained, the evaluation on single characters is computed. This is done by synthetically creating the characters from a to z in the same 20 fonts and the font size the OCR method had been trained with, to avoid additional bias. Hence 520 characters are classified into one of the 26 classes. If no interest points are detected at the cutting edge of a paper stripe the class #- is assigned.

To obtain the base performance of OCR, it is evaluated with the 520 complete character images. Thereafter horizontally and vertically cut character pieces are created and these 1040 images are then classified using OCR. In Table 1 the overall results of the evaluation can be seen. To retain an absolute sum of 520, every cut character piece is weighted by 0.5 and every complete character image with 1. Four tests have been performed:

case	#total	#tp	#fp	#-	precision
complete	520	500	20	0	0.9615
vertical cut	520	344	176	12	0.6615
horizontal cut	520	417	103	1.5	0.8019

Table 1. Comparison in classification performance

Complete single character evaluation To evaluate the base performance of the algorithm, the 520 single complete characters are classified, resulting in an overall precision of 0.9615, which can be seen in Table 1. The characters `a,t,j,l` are only classified incorrectly in 1 or 2 of the 20 cases, `n` is falsely predicted in 4 cases as an `m`. The weakest performance shows `i` which has an error rate of almost 50 %. This is consistent with the findings from [11] that characters like `i` and `j` for sans serif fonts exclusively produce SIFT features that represent corners with changing orientations and therefore are difficult to distinguish.

Vertically cut single character evaluation The precision of the vertical cut single characters is 0.6615. Vertical cuts are produced by shredding paper perpendicular to the word alignment which destroys the content of whole words and thus is more secure than shredding the paper horizontally. This time the character `i` is misclassified in 100% of the 40 cases and also `l` shows very poor recognition performance. In comparison to this `v` and `x` show perfect classification results, having a high precision of 0.9091 and 1 respectively.

Horizontally cut single character evaluation The horizontally cut single character evaluation showed better performance than the vertical one. A precision of 0.8019 was obtained for the 1040 cut character images. Except for `e` all characters resulted in similar (± 1) or better performance compared to the vertical cut evaluation.

Font evaluation Our test set of fonts contained 11 sans serif and 9 serif fonts. Best overall performance on all three scenar-

ios described above showed the serif font Garamond (0.9359), in comparison a result of 0.7436 is obtained by the sans serif font Arial. The average precision of the serif fonts was with 0.8426 higher than the result achieved by classifying the sans-serif font characters, which was 0.7803. This results from the fact that the OCR method is based on local features and serif fonts have characters with additional structural information.

4.2 Reconstruction of Shredded Documents

In this section pages are cut through arbitrary vertical planes producing equally sized strips. The same trained classifiers as in the former section were used.

In order to obtain paper strips with realistic cutting edges, pages with a size corresponding to an A4 page (approx. 2400 by 3400 pixels, font: Garamond, 35 lines, 300 dpi, font-size: approx. 18pt) were created from the first chapter of Daniel Defoe's Robinson Crusoe[1]. In order to make the matching harder, all pages have the same line spacing and number of lines.

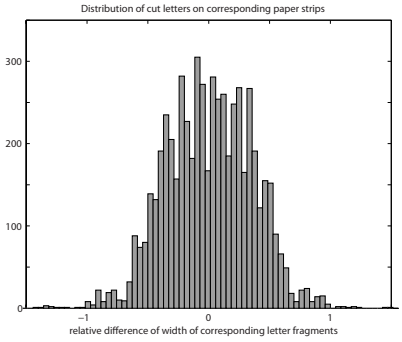
The resulting dataset consists of 13 pages, each of which is shredded into 20 fragments, producing 260 different fragments in total. Each page with its 20 strips therefore provides 19 left border segments and 19 right border segments, since the outer contours are not considered.

In order to find the distribution of cut-through characters the stripes were analyzed and the width of the characters cut on two adjacent fragments was compared. The average width of the characters (a–z) is 35 pixels, the widest character is m with 63 pixels, the narrowest characters are i, j and l with 19 pixels. Figure 6 shows the distribution of the cuts on the fragments ($\mu = 0.0063, \sigma = 0.3564$). The width of the cuts was subtracted and then divided by the width of the largest character, thus 0 represents characters which were cut vertically by their center. 17 % of the characters are cut within an interval of 5 pixels away from their center, 32 % within an interval of 10 pixels. Characters with an absolute width difference value bigger than 1 are caused by outliers which are due to connected characters or capital characters. An example can be seen in Figure 6(b). The connection between characters can be caused by serifs and a narrow distance.

Stripwise fixed matching The subsequent results were obtained by matching fixed size fragments onto each other. Thus, it is assumed that all fragments have the same size, which is a characteristic of strip shredded documents. Furthermore we presume that all fragments contain the same line information and that the orientation is known a priori. Within this work the line information was extracted by means of line histograms.

Stripwise fixed matching within the whole dataset In Table 2 the performance of matching one fragment border onto each opposite sided border can be seen. An increase in the number of lines on a fragment results in a higher precision, as the probability of cutting a larger number of characters and thus obtaining a larger number of probability histograms increases. This trend is visible in Figure 7 and Figure 8. Another effect

[1] www.fullbooks.com



(a)

(b)

Figure 6. (a) Distribution of character cuts within dataset, (b) connected characters

of a higher number of lines on the fragments is the decrease of the number of fragment borders which cannot be classified, shown by #- in Table 2. For full sized fragments (35 lines) in total 5764 cut characters are found, thus meaning that on every cutting edge in average about 23 characters are cut trough. From the 5764 characters solely in 2575 cases, probability histograms can be computed on both corresponding paper fragments. This means that from the 23 cut characters only about 10 characters produce histograms which can be used for matching. Matching the stripes using only their 5 first lines results in a precision of 0.0364. This is due to the fact that in this case for all 260 strips only 358 characters that are large enough for the OCR are found, which makes matching very difficult.

# lines	#tp	#fp	#-	precision
5	18	476	31	0.0364
10	37	457	16	0.0749
15	73	421	12	0.1478
20	98	396	9	0.1984
25	153	341	6	0.3097
30	201	293	3	0.4069
35	228	266	3	0.4615

Table 2. Statistics on whole dataset evaluation with mean as distance adjustion [n = 494 fragment borders]

Figure 7 shows the precision of matching within the whole dataset with different distance calculation methods. The best precision is achieved by calculating the mean of the single line distances. Using median or normalization of the probability histograms results in a significant decrease of the matching performance. The normalization generates smaller distances for correct matches if the characters are cut near their center. However, for characters that are cut near their edges, pieces produce a noisy histogram not clearly voting for any class. The normalization of these histograms then amplifies the noise and causes higher distances between the character pieces. Thus strips lacking matches at these locations can produce smaller global distances.

In addition to the evaluation of matching fragments onto each other in Figure 8 the precision when taking the n best

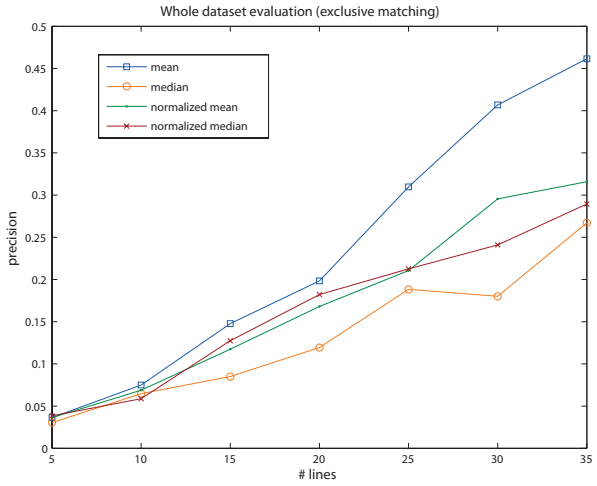


Figure 7. Matching performance on full dataset comparison for different number of lines on the fragments

matches into account can be seen. For full sized fragments (35 lines) a subset of the 5 best matches shows a precision of 77 %.

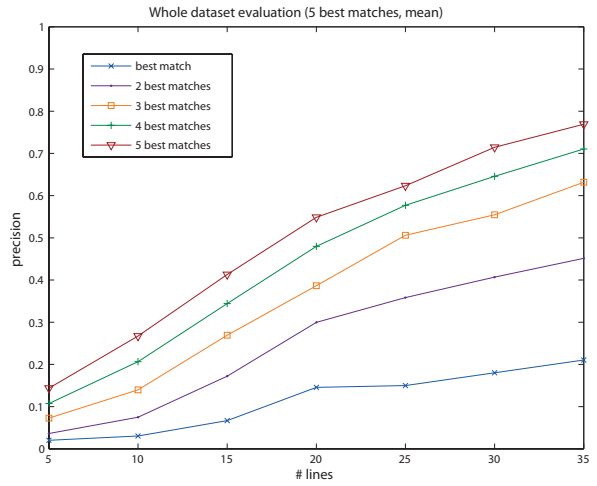


Figure 8. Matching performance taking the n best matches within the whole dataset

Stripwise fixed matching within single pages The evaluation solely within the strips of one page was done to investigate the performance on simulated (in this case perfect) clustering. The decrease of potential matches from over 250 to 19 results in a considerable increase in precision. Again a higher number of lines available increases precision. Figure 9 shows that for e.g. normalized median the performance of 20 line strips is better compared to the precision of 25 line strips. This is due to random optimization.

In Figure 10 the precision when taking up to the 3 best matches into account can be seen.

Matching of strips with different sizes Based on the scenario from Section 1 the following results show the ability of

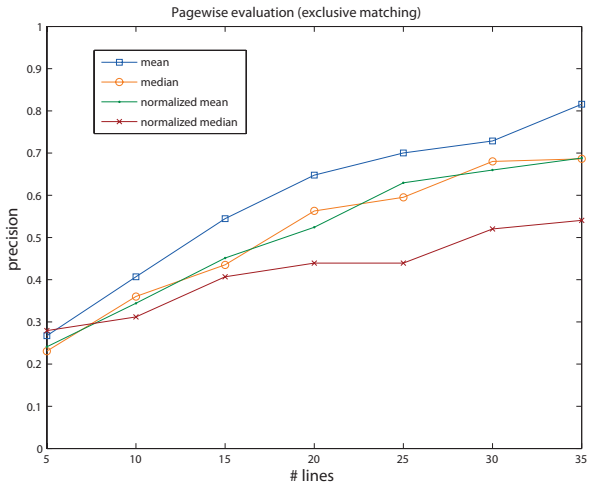


Figure 9. Matching performance on page comparison for different number of lines on the fragments

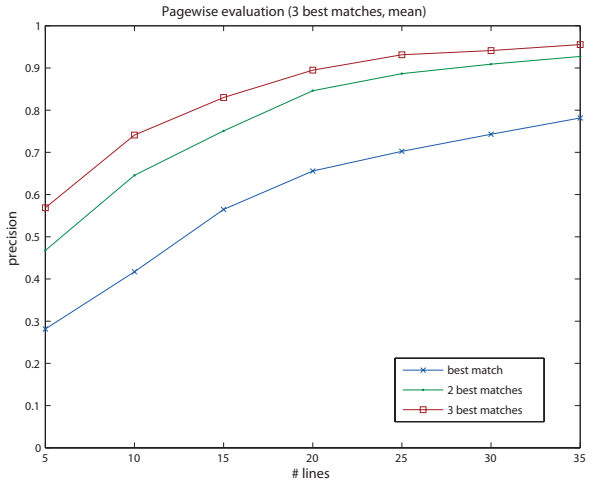


Figure 10. Matching performance taking the n best matches within the same page

the OCR algorithm to reconstruct two paper fragments of different size. The assumption is that two fragments belong together and can also be shifted vertically. Thus the positioning is not known a priori. Again all 260 full sized fragments (35 lines) were used. For every stripe we generate multiple matching fragments with different sizes of 5, 10, 15, 20 and 25 lines. Figure 11 shows the precision for fragments of the size of 25 lines matched onto full sized fragments. In contrary to the former results, the normalization increases the precision here. This is due to the size of the fragments, as with less lines available noisy histograms have a lower influence on the matching performance. To enhance the matching performance further the number of matching line positions is taken into account. As we presume that the fragments fully overlap each other, for every line which is matching between two stripes, a certain percentage of the total distance is removed, which is shown in the following figures on the abscissa.

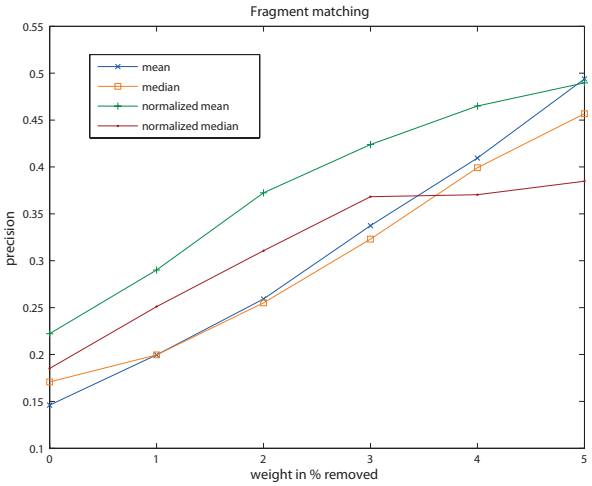


Figure 11. Matching precision for fragments of 25 lines matched onto 35 line fragments

In Figure 12 it can be seen that rewarding the matching lines does not increase the performance for strips containing only 5 lines. This is due to the fact, that the same percentage gets removed for a large number of positions, as much of them provide the same number of matching lines. However for bigger fragments the removal of distance concerning the matching lines results in a higher precision, as fewer constellations match perfectly.

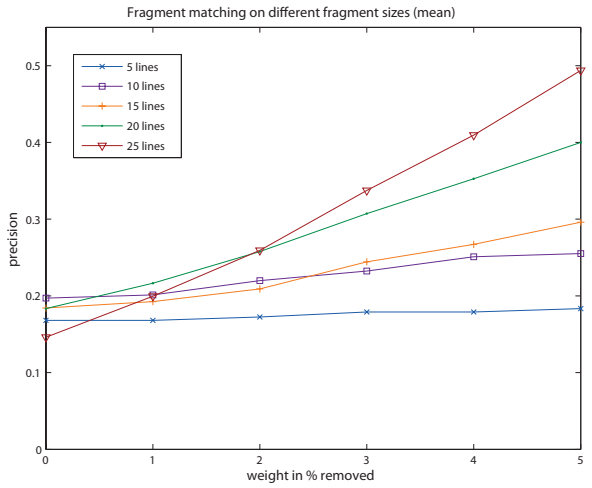


Figure 12. Matching performance for different sized strips with mean as distance calculation

5 Conclusion

In this work an approach that uses OCR to reconstruct shredded documents by classifying and reassembling cut characters in the border region of paper fragments was presented. The classification process is based on local descriptors computed at locations of interest points.

The OCR algorithm was evaluated on synthetic data in different scenarios in order to see its capability for document reconstruction. This work shows that OCR can be used to match cut characters in order to reassemble shredded paper stripes. As the proposed system was not intended to solve document reconstruction on its own, but rather to provide additional features, the OCR method should be combined with other approaches. A combination of the OCR with [3] can help to rule out wrong matching fragments by looking at the continuity of the matching character fragments. A clustering would additionally reduce the search space for the matching.

For future work it is highly desirable to identify the matching confidence of probability histograms. This allows to choose a more reliable subset of probability histograms. The number of interest points detected, the size or area of the cut character or a logarithmic normalization of the probability histograms are suitable criteria therefor.

References

[1] E. D. Demaine and M. L. Demaine, "Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity," *Graphs and Combinatorics*, vol. 23, no. 1, pp. 195–208, 2007.

[2] P. D. Smet, "Semi-automatic Forensic Reconstruction of Ripped-up Documents," in *Proceedings of ICDAR*, pp. 703–707, IEEE Computer Society, 2009.

[3] C. Schauer, M. Prandtstetter, and G. R. Raidl, "A Memetic Algorithm for Reconstructing Cross-Cut Shredded Text Documents," in *Hybrid Metaheuristics*, pp. 103–117, 2010.

[4] A. Ukovich and G. Ramponi, "Feature extraction and clustering for the computer-aided reconstruction of strip-cut shredded documents," *J. Electronic Imaging*, vol. 17, 2008.

[5] A. Ukovich, G. Ramponi, H. Doulaverakis, Y. Kompatsiaris, and M. Strintzis, "Shredded document reconstruction using MPEG-7 standard descriptors," in *Proceedings of the International Symposium on Signal Processing and Information Technology*, pp. 334–337, IEEE, 2004.

[6] A. Stieber, J. Schneider, B. Nickolay, and J. Krüger, "A Contour Matching Algorithm to Reconstruct Ruptured Documents," in *DAGM-Symposium*, pp. 121–130, 2010.

[7] A. Biswas, P. Bhowmick, and B. B. Bhattacharya, "Reconstruction of torn documents using contour maps.," in *Proceedings of ICIP-05, Vol. 3*, pp. 517–520, 2005.

[8] F. Kleber and R. Sablatnig, "A Survey of Techniques for Document and Archaeology Artefact Reconstructiong," in *Proceedings of ICDAR*, pp. 1061–1065, IEEE, 2009.

[9] H. C. Leitão and J. Stolfi, "Measuring the Information Content of Fracture Lines," *International Journal of Computer Vision*, vol. 65, pp. 163–174, 2005.

[10] M. Diem and R. Sablatnig, "Recognizing Characters of Ancient Manuscripts," in *Proceedings of SPIE*, vol. 7531, 2010.

[11] M. Diem and R. Sablatnig, "Are characters objects?," in *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 565 – 570, 2010.