

# **AUTOMATED MOSAICING OF TORN PAPER DOCUMENTS**

# 1 Introduction

Recovery of ripped-up documents is a problem that springs up in Archival study and Investigation science. Efficient and successful joining of torn pieces of papers to reconstruct the original documents is an important and challenging issue which if automated would alleviate the manual effort, which is difficult and time-consuming. This paper focuses on the reconstruction of documents ripped up by hand.

Document reconstruction problems can be solved in two steps, finding an initial set of matching fragment pairs as candidate matches followed by resolving the ambiguity among these candidate matches to reconstruct the original document.

Document fragments are represented by their contours and candidate matches are obtained by local curve matching of individual fragments. Local curve matching may result in many ambiguous matches and hence a search technique is required to resolve the ambiguities for global reconstruction. The best-first strategy is one such search technique which selects and merges the best matching pair to form a new fragment. This is repeated until there is only one fragment left which gives the recovered document.

Since document fragments often have similar contour segments in real applications, invalid matching occurs frequently using the local search strategies. Intensive backtracking in such cases makes these local search techniques inefficient. To overcome this, we resort to a global approach to disambiguate the candidate matches.

The global consistency approach is based on the fact although some incorrect matches may score higher than the correct one locally, the global score of the correct configuration will be much higher than that of an invalid one. A globally consistent solution is found by exploiting the relationships among all candidate matches.

## 1.1 Definitions, Acronyms and Abbreviations Used

**Turning Angle.** The turning angle at a certain point is an counterclockwise angle between the tangent of A at that point and the x-axis.

**Candidate Match Set.** A pair of fragments which can qualify to be a potential matchset in the future.

**Neighborhood.** Neighborhood is defined for a matchset which consists of the interactions of the given matchset with other matchsets

**Global Consistency.** The process of disambiguating candidate matches so as to reduce interactions between the various matchsets

**Fragments.** A part of an image is referred to as a fragment.

**Contour.** The outer boundary of the fragment is referred to as a contour.

## 1.2 Overview

Let  $F = \{F_1, F_2, \dots, F_L\}$  be the  $L$  fragments. The turning functions  $\{T_1, T_2, \dots, T_L\}$  be the turning functions of each fragment consisting of the turning angles of the points in the contour. The local matching between two fragments is calculated using their turning functions. The candidate matches put together form a match-set  $M = \{M_1, M_2, \dots, M_N\}$  having global match confidences  $x = \{x_1, x_2, \dots, x_N\}$ . The information about potential adjacency is applied for defining the neighbourhood  $G_i$  of a match  $M_i$ . The local interaction between neighbouring matches possibly propagates through the neighbourhood, to the entire set of candidates. The effect of the overall interaction is defined by global consistency  $C(x)$ .

The three major steps involved in the reconstruction are:

1. Mapping of the given fragments to contours.
2. Matching each of the contour segments, thereby generating a preliminary candidate match set.
3. The constituent matches of these sets are then disambiguated by a relaxation process.

For the third step, it is assumed that the candidate matches are well-connected so that interactions between neighbouring matches spread out to most of the candidates in the match-set. Without this assumption, we would not be checking for global consistency and the relaxation process will be ill-defined.

In real-life situations, conditions like the presence of fiber fringes along the tear lines and dissimilar fragments having similar edges need to be taken into consideration. They will be explained in the subsequent sections.

## 2 Finding Candidate Matches

### 2.1 Contour Representation

The contour is first extracted from the fragment and stored as a sequence of points in the x-y plane. This is followed by conversion of the above sequence to turning angles for efficient curve matching of two contours.

The turning angle at a certain point is the counter clockwise angle between the tangent of A at that point and the x-axis.

#### 2.1.1 Turning Function Computation Algorithm :

**Input :** Contour A

**Output :** Turning Function Vector T

**Steps :** For every point P in the contour

1. L is a vector of points of length delta to the left of P
2. R is a vector of points of length delta to the right of P
3. Calculate D, the angle difference between the L and R
4. Append D to the turning function sequence T

### 2.2 Partial Curve Matching

The turning angles of the contour points serve as a signature for the contour. So , Partial alignment of turning angles can be used as a method for finding matching fragment parts of the torn pieces. The partial sequence alignment is implemented using the Smith Waterman Dynamic Programming algorithm . Smith waterman algorithm, a widely used algorithm for pairwise DNA sequence alignment algorithm is slightly modified to fit in turning angle comparison using an optimized angle difference gap for contour comparison.

**Algorithm 2.2.1:**  $F(i, j)$

$$\text{MAX} \begin{cases} F(i-1, j) + gap \\ F(i, j-1) + gap \\ F(i-1, j-1) + match \text{ or } mismatch \text{ score} \\ 0 \end{cases}$$

The match, mismatch and the gap penalty values are optimized to get a nearly accurate match between the torn document fragments. The match between straight lines and turns have also been considered using a threshold value for the matching angles. The traceback of the maximum matched continuous turning angles is thus translated to contour points and is then used for merging the image fragments.

The algorithm for finding the maximally homologous turning angle subsequence is based on the algorithm used to find common molecular sub sequences. The related work in sequence matching is the homology algorithm by Needleman and Wunsch (1970), which first introduced the iterative matrix method of calculation. Other related works include heuristic algorithms by Fitch (1966) and Dayhoff (1969), Sankoff (1972), Reichert et al. (1973), Beyer et al. (1979) and Sellers (1974). The Sellers algorithm was generalized by Waterman et al. (1976) to include arbitrary insertions and deletions. Smith et al. (1980) shows that the generalized Sellers metric is equivalent to the homology algorithms suggested by Needleman and Wunsch (1970).

### 2.2.1 Partial Curve Matching Algorithm

**Input :** Contour A and Contour B

**Output :** Match Score s

**Steps :**

1. Initialize Matrix H to 0
2. Update H as follows:
  - a) If  $a_i$  and  $b_j$  are associated the similarity is  $H_{i-1,j-1} + s(a_i, b_j)$ .
  - b) If  $a_i$  is at the end of a deletion of length k the similarity is  $H_{i-k,j} - W_k$ .
  - c) If  $b_j$  is at the end of a deletion of length l the similarity is  $H_{i,j-l} - W_l$ .
  - d) A zero is included in place of calculated negative similarity, indicating that there is no similarity up to  $a_i$  and  $b_j$ .

The pair of segments with maximum similarity is found by first locating the maximum element of H. The other matrix elements leading to this maximum value are then sequentially determined with a traceback procedure ending with an element of H equal to zero. This procedure identifies the segments as well as produces the corresponding alignment. The pair of segments with the next best similarity is found by applying the traceback procedure to the second largest element of H and is not associated with the first traceback.

## 3 Global Consistency

For the complete reconstruction of the original object, a global technique is needed to eliminate the ambiguity resulting from local curve matching. The global solution is found by taking global consistency value and a global match confidence value for a given pair of fragments and then iterating the merging process of all fragments.

### 3.1 Procedure for computation and appropriate processing of global consistency :

**Input :** Candidate matches

**Output :** Global consistency value and global confidence values corresponding to the matches

**Steps :**

1. Obtain all possible candidate matches  $M_i$ , where  $M_i$  consists of matching fragments and match score.
2. Define a neighbourhood for each match  $M_i$ , consisting of all matches  $M_j$ , such that they have exactly one common fragment.
3. Define a compatibility parameter,  $ij$ , between matches  $M_i$  and  $M_j$  based on their overlap.
4. Initialize global matching confidence  $x_i$  for each  $M_i$ , based on match score
5. Compute global consistency  $C(x)$  using  $x$  and  $ij$ .
6. Update  $x$  values based on the trend of  $C(x)$ .
7. Compute new  $C(x)$  values using  $x$  values so obtained. Repeat the process till  $C(x)$  converges.
8. After converging, merge the matches with global confidence values 1.

## 4 Merging Fragments

Initially all the fragments are opened in a single global co-ordinate frame. As the matching is done using contour information only it is not required to handle and merge images in the intermediate steps. Instead just the contours and the images corresponding to the contours are maintained. This optimization reduces a lot of overhead. Initially Every fragment corresponds to one input image. As the algorithm proceeds fragments are merged by merging their contours. Every merging is characterised by 2 fragments say A and B, start point of A (startA), start point of B (startB), end point of A (endA) and end point of B (endB). To move the fragments homogeneous rigid transformation is used. In 2D plane every rigid transformation is characterised by 3 unique points sets. But as the curve matching algorithm may fail to give exact result, solving three simultaneous equations may lead to inconsistencies.

Instead all the points in between the start and end points are considered and a least square error minimization method is used to find an optimal transformation matrix align fragment B with fragment A. Also there may be slightly unequal number of points in the two match sets. So a uniform reservoir sampling from the larger one is used to make the match sets of equal size and it seems to give good results. If the transformation matrix is failed to be computed the merging is reported to be incompatible. Even if the transformation matrix is computed there may still be errors due to wrong matches.

So the overlapping area between A and transformed B is computed. If it is greater than a threshold (adjusted experimentally and also dependent on image sizes), the match is discarded. Otherwise it's a proper match, so B is transformed and aligned with A. Next optimal start and end points of the match set are found by travelling simultaneously on the contours and this also removes folds on the contours which may come while merging if not handled properly. In the fragments, the transformation matrices of all the image pieces which constitute them are also maintained. So while moving fragments the transformation matrices are also maintained properly. So at the end there will be a single fragment which will have all the image information and all the transformation matrices. So the images are loaded one by one, transformed, added pixelwise and the final merged image is produced.

### 4.1 Transformation Matrix Computation:

*Finds the transformation matrix to align a fragment along another fragment*

**Input :** Matchset

**Output :** Transformation Matrix

**Steps :**

1. If match size is not same in both the matches then make them of equal size by uniform reservoir sample of the larger one
2. Find the transformation matrix using OpenCV function `estimateRigidTransformation`
3. Return the matrix

## 4.2 Fragment Merging :

*Merges two given fragments along a match*

**Input :** Matchset containing Fragment A and B

**Output :** Merged Fragment

**Steps :**

1.  $T$  = Transformation Matrix for the matchset  $M$
2. Transform B with  $T$
3. Merge Contours of A and B and find optimal start and end points by travelling on the contours
4. The contour is stored in a new fragment F
5. Store all the images of A and B to F
6. Store all the transformation matrices of A in F
7. Left Multiply all the transformation matrices of B by  $T$  and store in F
8. Return F

## 4.3 Final Image Creation :

*Creates a final image by merging all the images of the final merged fragment*

**Input :** Fragment

**Output :** Merged Image

**Steps :**

1. Find the bounding box of the contour and ensure it to be in first quadrant by adjusting the transformation matrices
2. Also find the optimal height and width of the image
3. Load all the one by one , transform them by their corresponding transformation matrices and pixel wise add them
4. Return the final image



## 5 Results

Our software was tested with images of different nature. Images generated digitally, scanned images with shear and without shear were used. Number of fragments were varied and merged results were obtained as displayed below.

**Testcase with seven fragments:**

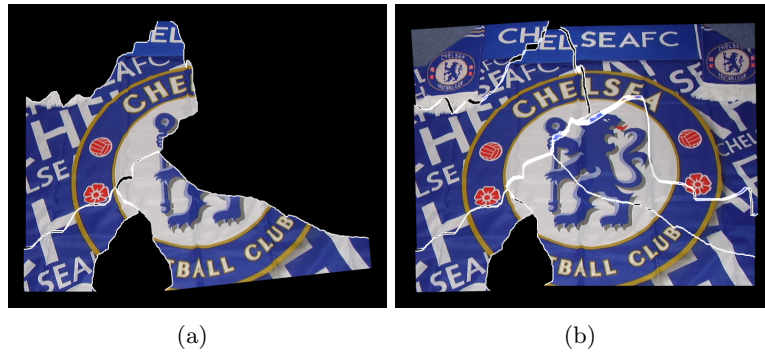


Figure 5.1: Intermediate Merging Steps



Figure 5.2: Final Merged Image

## Other Test Results



Figure 5.3: Two Fragments

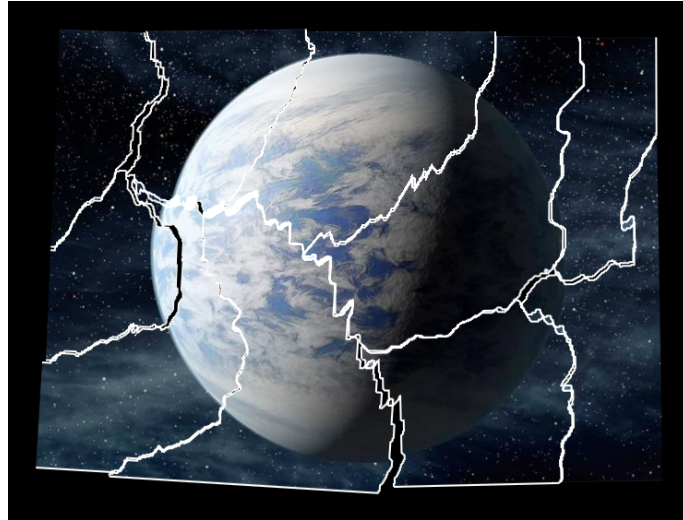


Figure 5.4: Ten Fragments



Figure 5.5: Ten Fragments(Running Time:150 seconds)

## 6 Discussions

Dynamic Programming algorithms (Smith Waterman) are slower and take up huge time and memory for computation of large image contour comparisons. In order to reduce the running time and efficiency of the software further improvements such as using heuristic methods would be helpful. Heuristic algorithms for sequence alignment which has been proved to be very successful in comparing DNA sequences might be a good substitute for aligning turning angles to obtain a faster match. Heuristic algorithm can be implemented by aligning shorter fragment parts and aligning them together with a huge gap penalty to obtain a faster local alignment for the whole fragment.

Apart from improving runtime by utilizing the heuristic approach, inclusions of few other criteria can boost the accuracy of the final output obtained. Employing the pixel matching score shall provide better results, especially during the presence of the almost straight line segments in the contours. After obtaining the candidate matches along with the segment of matching, computing the difference in the values of the pixels on either side of the segment of matching will provide another parameter, whose value shall provide another criterion for the matching accuracy. In the scenario of matches with straight line segments or contours with similar shapes, one can employ the pixel matching to disambiguate between the matches.

The algorithm, when made available on portable devices platform, can turn into a very user friendly application. The users can exploit the privilege of using the in-built camera to get the images of the torn fragments on-spot, and directly export it to the application for the matching process. Since use of heuristic approach reduces the computational load on the processor, recent and upcoming generation of handheld devices will be able to handle the application without any significant issue in the performance or the runtime.

# References

- [1] Biswas, P. Bhowmick, and B. B. Bhattacharya, Reconstruction of Torn Documents Using Contour Maps, In Proc. ICIP 2005, III:517-520.
- [2] L Zhu, Z Zhou and D Hu, Globally Consistent Reconstruction of Ripped-Up Documents, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 30 , Issue 1 (January 2008)
- [3] P. Faber, A Theoretical Framework for Relaxation Processes in Pattern Recognition: Application to Robust Nonparametric Contour Generalization, IEEE Trans. Pattern Analysis and Machine
- [4] C. Solana, E. Justino, L.S. Oliveira, and F. Bortolozzi, Document Reconstruction Based on Feature Matching, Proc. 18th Brazilian Symp. Computer Graphics and Image Processing, pp. 163-170, 2005
- [5] J. McBride and B.B. Kimia, Archaeological Fragment Reassembly Using Curve-Matching, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1-8, 2003.
- [6] O. Faugeras and M. Berth, Improving Consistency and Reducing Ambiguity in Stochastic Labeling: An Optimization Approach, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 3, no. 4, pp. 412-424, 1981. Intelligence, vol. 25, no. 8, pp. 1021-1027, Aug. 2003.