Simultaneous Reconstruction of Multiple Hand Shredded Content-Less Pages Using Graph-Based Global Reassembly

K.S. Lalitha[1(✉)], Sukhendu Das[1], Arun Menon[2], and Koshy Varghese[2]

[1] Department of Computer Science and Engineering, IIT Madras, Chennai, India
kslalitha@cse.iitm.ac.in, sdas@iitm.ac.in
[2] Department of Civil Engineering, IIT Madras, Chennai, India
{arunmenon,koshy}@iitm.ac.in

Abstract. Hand shredded content-less pages reassembly is a challenging task. This has applications in forensics and fun games. The process is even more tedious when the number of pages from which the fragments are obtained is unknown. An iterative framework to solve the jigsaw puzzles of multiple hand shredded content-less pages has been proposed in this paper. This framework makes use of the shape-based information alone to solve the puzzle. All pairs of fragments are matched using the normalized shape-based features. Then, incorrect matches between the fragments are pruned using three scores that measure the goodness of the alignment. Finally, a graph-based technique is used to densely arrange the fragments for the global reassembly of the page(s). Experimental evaluation of our proposed framework on an annotated dataset of shredded documents shows the efficiency in the reconstruction of multiple content-less pages from arbitrarily torn fragments and performance metrics have been proposed to numerically evaluate the reassembly.

Keywords: Content-less page reassembly · Partial contour matching · Shape features · Agglomerative Clustering · Multiple page reassembly

1 Introduction

Environmental conditions, like fire and weathering, and human activities cause damages to the paper documents. These may result in the shredding of the paper documents. In fields like forensics, gaming, and archaeology, deciphering the content in shredded paper documents is a task of great importance. The process becomes even more complex when the shredded fragments have no content in it and the number of documents from which the fragments are obtained is unknown. Typically, features of fragments based on shape, color, texture or combinations of these, are used in reassembly. We solve the problem of reassembly using the shape (contour) information alone as the apictorial puzzles have no content in them. In this paper, we propose an approach to automatically reassemble hand shredded fragments from multiple content-less pages and we also propose performance metrics to measure the quality of the reassembly.

S. Mukherjee et al. (Eds.): ICVGIP 2016, LNCS 10481, pp. 75–87, 2017.
https://doi.org/10.1007/978-3-319-68124-5_7

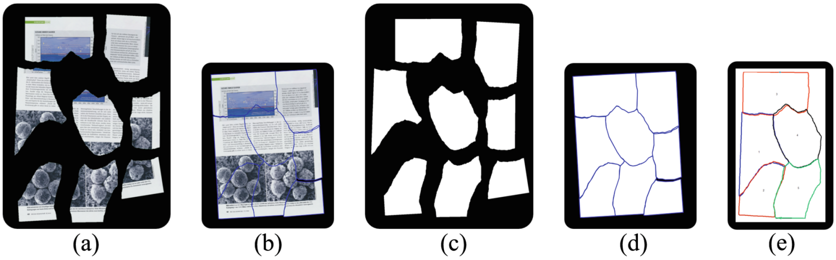


Fig. 1. Example page from dataset [11]. (a) Eight hand shredded paper fragments with content. (b) Reassembled page with content (ground-truth). (c) Binary masks (content-less) of the paper fragments, used as input for our problem. (d) Page reassembled using the binary masks in (c), by our proposed method. (e) Result generated by the method proposed in [6], when the features are extracted from the smoothed contours of the fragments in (c).

In our work, we use the *bdw082010* dataset [11], which has 48 double sided sheets, shredded by hand into 8 fragments each. Since, these sheets have information on both front and back sides, the dataset contains 96 pages. The fragments in this dataset have almost arbitrary shapes. For reassembly, only the binary segmentation mask of each of the fragments is used. Figure 1(a–b) shows eight fragments (with contents) from a page and the original document itself. Figure 1(c) shows the binary masks of the fragments. Our work reassembles the eight fragments in Fig. 1(c) into Fig. 1(d). Figure 1(e) shows the failure of method proposed in [6], one of the state of art methods to solve apictorial jigsaw puzzle.

The main contributions of our work are as follows:

1. We propose performance metrics to evaluate the reassembly numerically.
2. We propose an efficient framework to reassemble multiple content-less pages using graph-based clustering algorithm.
3. Our approach is unsupervised, that is, it does not require any prior information about the final reassembled shape and the number of pages to which the input fragments belongs to.

The rest of the paper is organized as follows: Sect. 2 discusses the related work. Section 3 gives a detailed overview of the framework proposed for reassembly of hand shredded content-less pages. Section 4 describes the experimental results. Section 5 concludes the work.

2 Related Works

The problem of reassembly of fragmented sheets is solved by considering shape-related features in [4–8,10,16,19], color/texture-related features in [14,17] and combination of shape-related & color-related features in [9,11,18]. In some works, as in [16], the contours are divided into contour segments, delimited by the corners, which are matched. Generally, distance metrics are used to measure the similarity of the features of the boundary points. In [11], Richter *et al.* proposed a supervised approach using an SVM classifier, to find points in the boundary with similar features. Most of the recent methods rely on texture features in the fragments to align globally and locally.

Global reassembly algorithms use a greedy approach to iteratively merge fragments. In [2], methods for global reassembly includes global relaxation to reduce the error in the final reconstruction. In [18], a graph-based algorithm that performs better groupwise matching and a variational graph optimization to minimize the error in final reconstruction are proposed. In [19], the candidate disambiguation problem has been formulated to define the compatibility between the neighboring matches and the global consistency is defined as the global criterion. In [9], Liu *et al.* proposed a spectral clustering based approach to reassemble multiple photos simultaneously.

The algorithms developed in [4–6,8,10] solve apictorial jigsaw puzzles. In [5], the "indents" and "outdents" are matched using ellipses fitted into them. In [10], the curve fitting is done using polar coordinate systems centered around the local extrema of the curvature. In [8], dynamic programming methods are used to match the curvature and arc length invariants. Once the matches are known, the robust relative transformations between the fragments are estimated using variants of ICP [13] or variants of MLESAC [12]. Incorrect matches are eliminated by locally verifying the transformation between fragments as in [12] or by checking the global compatibility between matches as in [18].

Recent work in [6], efficiently solves apictorial jigsaw puzzles. Here, the contours are decomposed into smaller arcs by *bivertex decomposition* and the arcs are then matched to fit the fragments. Also, an efficient algorithm to minimize the local error in alignment of fragments is proposed. However, as shown in 1(e), the method does not reassemble the hand shredded pages, as this method does not consider the overlap between fragments and the global placement of fragments during the reassembly. Our paper bridges the gap by reassembling shredded content-less pages.

3 Reassembly Framework

The main aim of the paper is to reassemble multiple hand shredded content-less pages using shape information alone. The iterative framework, shown in Fig. 2, is used in the reassembly process. The stages are discussed below:

3.1 Feature Extraction

The contours of the input fragments are approximated using the Douglas-Peucker algorithm [3], with parameter σ, to reduce the processing complexity. Given N input fragments from the shredded pages, $\widehat{\mathcal{U}} = \{U^1, U^2, \ldots, U^N\}$ and $\widehat{\mathcal{V}} = \{V^1, V^2, \ldots, V^N\}$ are the sets containing the actual contours and approximated

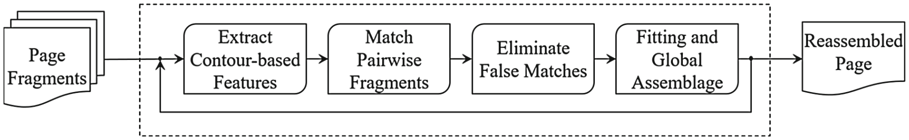


Fig. 2. The proposed iterative framework with various processing stages, for hand shredded content-less page reassembly.

contours, respectively. Each $V^i = \{v^i_1,\ v^i_2,\ \ldots,\ v^i_{n_i}\} \subseteq U^i$ describes the i^{th} fragment, where, n_i is the number of points in the approximated i^{th} contour.

The following shape-based features are extracted from the vertices of the approximated contours:

- *Log of Absolute value of Signed Curvature* (κ) - assume, fragments' boundaries are *Jordan curves* of class C^2.
- *Log of Mean of Edge Lengths* (μ), from current vertex to the two adjacent vertices.
- *Angle between two vectors* (θ), one joining previous vertex & current vertex and other joining current vertex & next vertex.

The skewness measures of the distributions of the above features are close to 0, resulting in a equally weighted 3-dimensional feature vector at a vertex, $f = [\kappa, \mu, \theta]$. *Min-Max Normalization* is used to normalize features. The matrices containing the normalized feature vectors of the i^{th} fragment are $FV^i_c, FV^i_a \in \mathbb{R}^{n_i \times 3}$, obtained by traversing in clockwise and anti-clockwise direction, respectively.

3.2 Pairwise Matching of Fragments

In this phase, all the possible alignment between two fragments are found out and the transformation corresponding to the alignments are estimated. Then, three scores are computed to evaluate the goodness of each of the alignments numerically. The process is repeated for all the pairs of fragments.

Similar Contour Segments Discovery. Using a modified version of Smith-Waterman (SW) algorithm [15], similar contour segments between two randomly chosen fragments, say i and j, are found out. A feature vector in FV^i_c is considered to be similar to another feature vector in FV^j_a, if the Euclidean distance between the feature vectors is less than ζ. Based on this, the common subsequences between FV^i_c and FV^j_a are found out. If M is the number of common sub-sequences between FV^i_c and FV^j_a, then there M possible ways in which fragments i and j can be aligned. For every $m \in 1, \ldots, M$, let $sp^i_m \subseteq V^i$ and $sp^j_m \subseteq V^j$ denote the sets containing contour points in fragment i that are similar to contour points in fragment j. Thus, the set $\mathbf{SP}^{i,j} = \{\{sp^i_1, sp^j_1\}, \{sp^i_2, sp^j_2\}, \ldots, \{sp^i_M, sp^j_M\}\}$ contains all the matched contour segments pairs. Matching using FV_a^i and FV_c^j also gives rise to the same set of matches, but the ordering (traversal direction) of points in the contour segments are reversed.

Estimation of Transformation and Scores. For every matched contour segments pair in $\mathbf{SP}^{i,j}$, a transformation is estimated. Unlike the prior works [12,13], we find a pair of points α, $\alpha' \in sp_m^i$ and the corresponding points $\beta, \beta' \in sp_m^j$, such that $|\text{Len}(\alpha, \alpha') - \text{Len}(\beta, \beta')| \leq \delta$, where $\text{Len}(\alpha, \alpha')$ is the arc length between α and α', and δ is a parameter. A translation vector, $\boldsymbol{\lambda}$, is computed from the offset between α and β, as $\boldsymbol{\lambda} = \alpha - \beta$. The rotation angle, ϕ, is the angle between vectors $\overrightarrow{\alpha\alpha'}$ and $\overrightarrow{\beta\beta'}$. The Euclidean transformation, $A_m^{i,j} \in \text{SE}(2)$, applied to fragment j to align it with fragment i, is thus represented using its parameters as $A_m^{i,j} = (\phi, \boldsymbol{\lambda}) \in S^1 \times \mathbb{R}^2$ and the set containing the transformations between i and j is $\mathbf{A}^{i,j} = \{A_1^{i,j}, A_2^{i,j}, \ldots, A_M^{i,j}\}$. For each $A_m^{i,j} \in \mathbf{A}^{i,j}$, we calculate three scores, which are commonly used as topological features for 2-D shapes:

1. ***Connectivity*** *between fragments* is the ratio of the length of the common sub-sequence between the fragments to the minimum of the two perimeters.

$$CNS_m^{i,j} = \frac{\text{Len}(sp_m^i)}{\min(\text{Perimeter}(V^i), \text{Perimeter}(V^j))}, \begin{cases} i, j \in 1, \ldots, N, \ i \neq j, \\ m = 1, \ldots, M. \end{cases} \tag{1}$$

2. *Relative* ***Fitness*** *value* is the ratio of the sum of the areas of overlap and gap between fragments to the mean of areas of the two fragments.

$$\begin{aligned} FTS_m^{i,j} &= 2\frac{|OA^{i,j}| + |GA^{i,j}|}{\text{Area}(V^i) + \text{Area}(V^j)}, \begin{cases} i, j \in 1, \ldots, N, \ i \neq j, \\ m = 1, \ldots, M. \end{cases}, \\ \text{where, } OA^{i,j} &= \text{Area}(V^i \cap AV^j), \\ GA^{i,j} &= \text{Area}(V_m^{i,j}) - \text{Area}(V^i \cup AV^j), \\ V_m^{i,j} &= \text{Boundary}(V^i \cup AV^j), \text{ where } AV^j = A_m^{i,j} \star V^j. \end{aligned} \tag{2}$$

   Operator '$\star$' denotes the transformation for all points in a fragment.

3. ***Compactness*** *of the merged Fragment* is the ratio of the square of the perimeter of the merged fragment to the area of the merged fragment.

$$CMS_m^{i,j} = \frac{[\text{Perimeter}(V_m^{i,j})]^2}{\text{Area}(V_m^{i,j})}, \begin{cases} i, j \in 1, \ldots, N, \ i \neq j, \\ m = 1, \ldots, M. \end{cases}. \tag{3}$$

The longer the common sub-sequence between the fragments, the better is the alignment. Thus, the *Connectivity* score should have larger values. The lesser the area of overlap and gap between the fragments, the better is the alignment. Thus, the *Fitness* score should have smaller values. The denser the fragments are arranged, the better is the alignment. Thus, the *Compactness* score should also have smaller values.

3.3 Elimination of Incorrect Matches and Graph Generation

This phase eliminates the incorrect matches between two fragments based on the associated scores and finds the best possible alignment between two fragments. This is a four step process. First, we eliminate the matches that have *Connectivity* scores less than th_{CON} and *Fitness* scores greater than th_{FIT}, and the remaining transformations form a set of $\widetilde{\mathbf{A}}^{i,j} \subseteq \mathbf{A}^{i,j}$, $\forall i, j \in 1, \ldots, N$ and $i \neq j$, such that the cardinality of the set is $\left|\widetilde{\mathbf{A}}^{i,j}\right| = M' \leq M$.

Then, the Locking algorithm proposed in [6] is used to reduce the errors in the transformations. The parameters values of $K_1, \ldots, K_4$, ϵ, ν, ρ and j_{max} used are same as that given in [6]. Let, $\widetilde{\mathbf{LA}}^{i,j} = \text{Locking}(\widetilde{\mathbf{A}}^{i,j})$, $\left|\widetilde{\mathbf{LA}}^{i,j}\right| = M'$, be the set of error-corrected transformations. Application of this algorithm leads to increase in the *Connectivity* score and decrease in the *Fitness* score (recomputed).

Then, based on the *Connectivity* score, we again eliminate transformations and form a set $\widetilde{\mathbf{BA}}^{i,j} = \{\widetilde{BA}_1^{i,j}, \ldots, \widetilde{BA}_{M''}^{i,j}\} \subseteq \widetilde{\mathbf{LA}}^{i,j}$ and $\left|\widetilde{\mathbf{BA}}^{i,j}\right| = M'' \leq M'$.

Finally, Single Best Transformation that align j with i, $SBA^{i,j}$, is estimated based on the *Compactness* (Cms) score of the transformations in $\widetilde{\mathbf{BA}}^{i,j}$ as:

$$SBA^{i,j} = \begin{cases} \underset{\substack{\widetilde{BA}_k^{i,j} \\ k=1,2,\ldots,M''}}{\arg\min} \ \text{Cms}(\widetilde{BA}_k^{i,j}), & \text{if } \left|\widetilde{\mathbf{BA}}^{i,j}\right| \neq 0, \\ null, & \text{Otherwise} \end{cases}, \quad \begin{matrix} \forall i, j \in 1, \ldots, N, \\ i \neq j. \end{matrix} \tag{4}$$

From Eq. (4), the single best transformation to align i with j is:

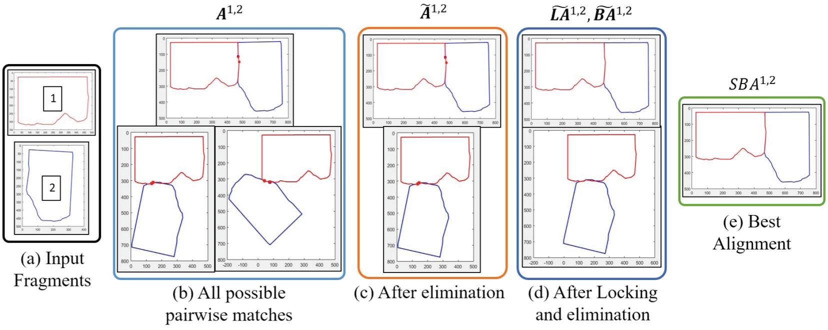
$$SBA^{j,i} = \begin{cases} (SBA^{i,j})^{-1}, & \text{if } SBA^{i,j} \neq null \\ null, & \text{Otherwise} \end{cases}, \quad \begin{matrix} \forall i, j \in 1, \ldots, N, \\ i \neq j. \end{matrix} \tag{5}$$

The above two transformations are appended to the set $\mathbf{SBA}$, containing all the Single Best Transformations, $\mathbf{SBA} = \mathbf{SBA} \cup \{SBA^{i,j}, SBA^{j,i}\}$. If $SBA^{i,j} \neq null$ and $SBA^{j,i} \neq null$, we then add an edge, e_{ij}, between nodes i and j in the undirected graph, $G(\mathbf{V}, \mathbf{E})$, formed with input fragments as its nodes. Weight of the edge e_{ij} is:

$$w(e_{ij}) = \text{Cms}(SBA^{i,j}), \qquad i, j \in 1, \ldots, N, \ i \neq j. \tag{6}$$

Figure 3(a)–(e) shows an example of the process of elimination of incorrect pairwise matches of fragments. The final match in Fig. 3(e) is the best of all the matches identified by the Pairwise Fragment Matching phase in Fig. 3(b). Locking reduces the error in the alignment. The above steps are applied to all the possible pairings.

To increase the robustness during multiple pages reassembly, top 70 to 100 percent of the best matches are alone considered to be the input for the global reassembly phase.


$\boldsymbol{A}^{1,2}$
$\widetilde{\boldsymbol{A}}^{1,2}$
$\widetilde{\boldsymbol{LA}}^{1,2}, \widetilde{\boldsymbol{BA}}^{1,2}$
$SBA^{1,2}$
1
2

(a) Input Fragments

(b) All possible pairwise matches

(c) After elimination

(d) After Locking and elimination

(e) Best Alignment

Fig. 3. Elimination of incorrect matches between fragments 1 and 2, for the example shown in Fig. 1(c). The fragment indices are given in rectangular boxes. (a) Pair of input fragment contours 1 and 2. (b) All possible pairwise matches between the fragments. (c) Pruned matches after elimination, based on transformation scores (Eqs. (1)–(3)). (d) Matches retained after locking the fragments and again eliminating matches based on transformation scores. (e) Best alignment between the pair.

3.4 Global Reassembly

In this phase, transformations are applied to approximately align the fragments. Unlike other methods that start from a seed fragment and then greedily aligns other fragments, the method proposed in this paper follows a Modified Agglomerative Clustering Algorithm to align the fragments together. Initially, consider each fragment as a cluster. Then, detect clusters corresponding to the best alignment, based on the edge weights of the graph G. Here, lesser the edge weight better is the alignment. Merge the pair of clusters, if the alignment of fragments in the participating clusters do not lead to overlap of fragments in the reassembled page. Choose the next best alignment and repeat the steps until there are no more clusters that can be merged without overlapping.

When fragments are clustered together, it implies that fragments can be assembled without any overlap, by applying the appropriate transformations. The proposed method for global reassembly is similar to an algorithm proposed in [11]. However, in the proposed method, the weight matrix need not be updated after every iteration. This reduces the computational overhead significantly.

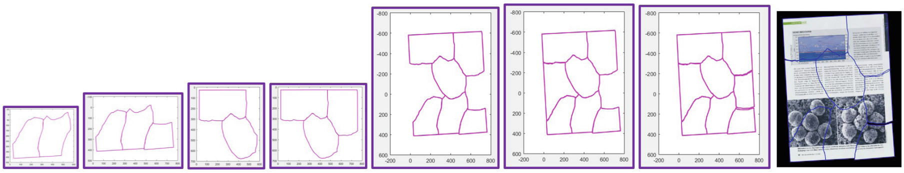
The steps of the proposed method for global reassembly are given in Algorithm 1. We start by considering N singleton clusters as an initial set of clusters $\mathcal{C}^{(0)} = \{c_1, \dots, c_N\}$. Each cluster contains a fragment index, $c_i = \{i\}$, $i \in 1, \dots, N$. The transformations to be applied to the fragments corresponding to the elements of cluster c_k is stored in the set $\tau_k \in \mathrm{SE}(2)$. Initialize $\tau_k = \{I\}, \forall k = 1, \dots, N$, where $I = (0, 0, 0) \in S^1 \times \mathbb{R}^2$ represents the identity transformation. Define a set containing all the initial transformations of the clusters as $\Gamma^{(0)} = \{\tau_1, \dots, \tau_N\}$. The sets $\mathcal{C}^{(0)}$ and $\Gamma^{(0)}$, along with set of fragments $(\widehat{\mathcal{U}})$, graph (G) and set of single best transformations (**SBA**), are input to Algorithm 1.

Algorithm 1. *Algorithm for global reassembly:* initial set of fragments are given as input. Output contains clusters of fragments, that are aligned together without overlap, and their transformations.

```
Input: $\widehat{\mathcal{U}} = \{U^1, \ldots, U^N\}$: Set of fragments, $G(\mathbf{V}, \mathbf{E})$: Graph,
       $\mathcal{C}^{(0)}$: Initial set of clusters, $\Gamma^{(0)}$: Initial set of cluster transformations,
       SBA: Set of Single Best Transformations
Output: $\mathcal{C}^{(size(\mathbf{E}))}$: Final set of clusters,
        $\Gamma^{(size(\mathbf{E}))}$: Final set of cluster transformations

Sort $\mathbf{E}$ in increasing order of weights
Set $iter \leftarrow 1$
while $iter \leq$ size($\mathbf{E}$) do
    Assign $a$ and $b$ to the nodes connected by edge $\mathbf{E}(iter)$
    Find the clusters $c_A$ and $c_B$ to which $a$ and $b$ belong to, respectively
    Set $flag \leftarrow 0$
    if $c_A$ and $c_B$ are not same then
        $(\widehat{c_A}, \widehat{\tau_A}) \leftarrow$ NewCluster($a, b, c_A, c_B, \tau_A, \tau_B, SBA^{a,b}$)  /* See Algorithm 2 */
        if fragments in $\widehat{c_A}$ do not overlap then
            $\mathcal{C}^{(t)} = \{\mathcal{C}^{(t-1)} \setminus \{c_A, c_B\}\} \cup \widehat{c_A}$; $\Gamma^{(t)} = \{\Gamma^{(t-1)} \setminus \{\tau_A, \tau_B\}\} \cup \widehat{\tau_A}$
            Set $flag \leftarrow 1$
        end
    end
    if $flag = 0$ then
        $\mathcal{C}^{(t)} = \mathcal{C}^{(t-1)}$; $\Gamma^{(t)} = \Gamma^{(t-1)}$
    end
    Increment $iter$ by 1
end
return $\left(\mathcal{C}^{(size(\boldsymbol{E}))}, \; \Gamma^{(size(\boldsymbol{E}))}\right)$
```

Algorithm 2 shows the steps of the function used to combine clusters and to compute the transformations for elements in the new cluster. Figure 4 shows examples of new valid clusters formed at the end of Algorithm 1. Figure 4(h) shows the page reassembled by the proposed method.



(a) iter = 1 (b) iter = 2 (c) iter = 3 (d) iter = 4 (e) iter = 5 (f) iter = 6 (g) iter = 7 (h) Final Result

Fig. 4. (a)–(g) New clusters formed at the end of every iteration in the global reassembly phase, for the example shown in Fig. 1(c). Here, *iter* indicates the iteration number in Algorithm 1. (h) The final reassembly result with content.

Algorithm 2. *Function to combine clusters:* two clusters (c_A, c_B) along with their transformations (τ_A, τ_B) are the input. Output is a new cluster $(\widehat{c_A})$ formed by combining the input clusters and transformations $(\widehat{\tau_A})$ to be applied to the elements in the new cluster.

```
Function NewCluster($a, b, c_A, c_B, \tau_A, \tau_B, SBA^{a,b}$)
    New cluster, $\widehat{c_A} = \{c_A \cup \{b\} \cup \{c_B \setminus \{b\}\}\}$
    Initialize $\widehat{\tau_A} = \emptyset$
    for each $l$ in $c_A$ do
        $\widehat{\tau_A} = \widehat{\tau_A} \cup \{\text{tran}(A, l)\}$
        /* tran$(A, l) \in \tau_A$ represents the transformation applied to
           fragment corresponding to element $l$ in cluster $c_A$ */
    end
    Append transformation for element $b$ as $\widehat{\tau_A} = \widehat{\tau_A} \cup \{\text{tran}(A, a) * SBA^{a,b}\}$
        /* Operator $'*'$ denotes multiplication of transformations */
    $temp = (\text{tran}(A, a) * SBA^{a,b}) * \text{Inverse}(\text{tran}(B, b))$
    for each $l$ in $\{c_B \setminus \{b\}\}$ do
        $\widehat{\tau_A} = \widehat{\tau_A} \cup \{temp * \text{tran}(B, l)\}$
    end
return $(\widehat{c_A}, \widehat{\tau_A})$
```

4 Experiments and Results

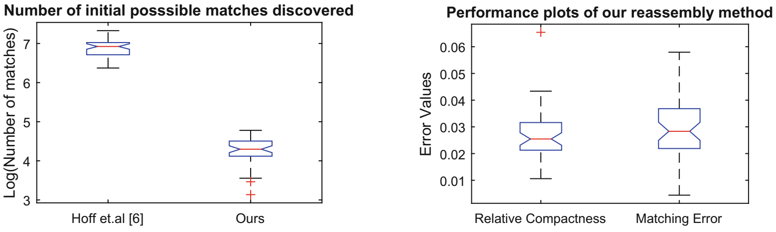
Experiments are done to evaluate the reassembly accuracy numerically and to evaluate the contributions of the method in reassembling multiple documents.

The empirical values of the parameters used in all the experiments are follows:

$$\sigma : 1.4\text{–}3.1, \quad \zeta : 0.06\text{–}0.10, \quad \epsilon = 0.0001, \quad \nu = 3,$$
$$K_1 = 15, \quad K_2 = 4, \quad K_3 = 1, \quad K_4 = 0.5, \quad \delta = 5,$$
$$th_{CON} : 0.10\text{–}0.15, \quad th_{FIT} : 0.009\text{–}0.03, \quad \rho = 1/3, \quad j_{max} = 50.$$

The performance of the proposed algorithm solely depends on the choice of the above parameters. The reassembly approach discussed reassembles all the 96 distinct sheets available from the dataset [11] in at most 2 iterations.

Figure 5(a) shows that the *bivertex decomposition* method, proposed in [6] to find the initial set of possible matches, returns a larger set of hypotheses than our proposed method. Thus, the running time of the proposed method is comparatively less. According to [1], the worst-case lower bound on the number of iterations performed by ICP algorithm in order to converge is $\Omega(n/d)^{d+1}$, where n is the size of the input data point set and d is the dimensionality of the input data. In this work $d = 2$ and thus, if ICP algorithm is used to estimate the transformation, given n pairs of matched points, the worst-case lower bound on the number of iterations performed by ICP algorithm in order to converge is $\Omega(n^3)$. However, the running time complexity of the method proposed in this paper, in Pairwise Matching of Fragments phase, to estimate the transformation



(a)

(b)

Fig. 5. (a) Number of initial possible matches discovered. (b) Performance plots of our reassembly method showing the distribution of the two error values. Box plots shows the median value (red line) as well as the 25th and 75th percentiles. The '+' symbols indicate the outliers (Color figure online).

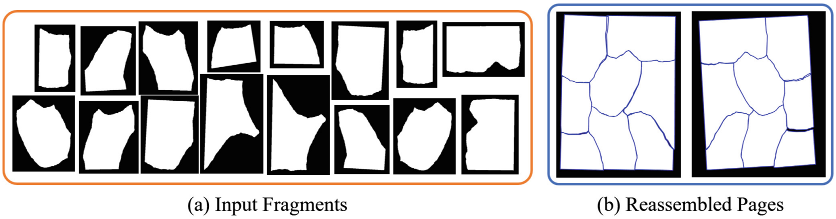
is $O(n^2)$. Hence, the proposed work is much faster than the prior works, in estimating the transformation.

Performance Evaluation: To evaluate the performance of the reassembly framework numerically, we use the following two error measures:

1. ***Relative Compactness Error:*** It is the ratio of the difference in the *Compactness* scores of the final reassembled page and the minimum boundary rectangle, which encompasses the final reassembled page, to the *Compactness* score of the final reassembled page. Since in our case we know that the fragments should be reassembled into a rectangle, we are comparing the compactness score of the reassembly output of our method with the compactness score of the minimum bounding rectangle, and using the relative difference value as a performance metric.
2. ***Matching Error:*** This error measure, proposed in [18], is the ratio of the average distance between matching segments of the fragments to the diagonal of the minimum bounding rectangle encompassing the reassembled page.

Figure 5(b) shows the box plots of the two error values obtained by our method for the 96 distinct pages. It can be seen from the plot that the maximum value of the *Matching Error* obtained using our method is around 0.06, which is at least 40 percent less than the error values of the reassembly approach proposed in [18]. The least *Matching Error* value reported in [18] is 0.10.

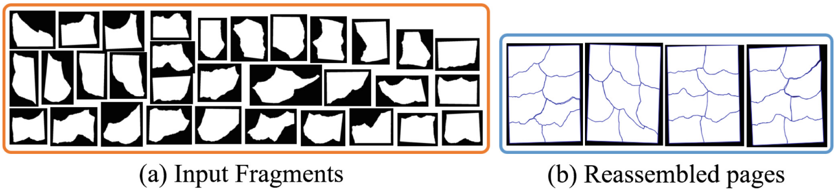
Multiple Pages Reassembly: In a content-less fragment, generally, it is hard to find which side (front or back) of the fragment is to be used in the reassembly process. We evaluate our method by inputting both sides of the fragments simultaneously. Figure 6 shows the reconstruction of one such sheet. It is observed that the method is capable of reconstructing both front and back sides of the sheet simultaneously.



(a) Input Fragments (b) Reassembled Pages

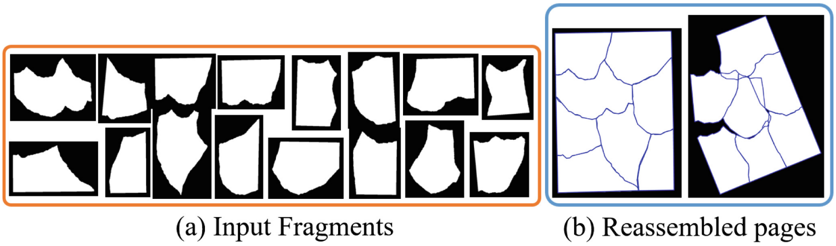
Fig. 6. Simultaneous reassembly of front and back of sheet 2, from dataset [11]. (a) Input: 8 fragments, both front and back sides scans. (b) Reassembled pages.

Simultaneous reassembly of fragments from multiple pages is challenging. When the number of sheets that has been shredded is unknown, the task becomes even more difficult. Figure 7(b) shows the result of the proposed method, when the 32 input fragments, shown in Fig. 7(a), are from four different pages. The method took 2 iterations to give output shown in Fig. 7(b). Figure 8(b) shows the result of one failure case of the proposed method, for the input in Fig. 8(a). The failure is due to an error in the matching of fragments.



(a) Input Fragments (b) Reassembled pages

Fig. 7. Simultaneous reassembly of four pages, from dataset [11]. (a) 32 input fragments. (b) Four correctly reassembled pages.



(a) Input Fragments (b) Reassembled pages

Fig. 8. Simultaneous reassembly of two pages, from dataset [11]. (a) 16 input fragments. (b) Two reassembled pages, with one failure case.

5 Conclusion

A novel iterative framework for shape features based automatic reassembly of multiple hand shredded content-less pages has been proposed. Experiments are done to show the effectiveness of the proposed approach. The reassembly has been evaluated numerically using the performance metrics and experiments show that our method has less error than the existing approaches.

In future work, the difficulties introduced due to material loss will be handled. Exploration for extending the framework to reassemble 3-D broken objects would be an appropriate scope of future work.

References

1. Arthur, D., Vassilvitskii, S.: Worst-case and Smoothed Analysis of the ICP algorithm, with an application to the K-means method. In: 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 153–164 (2006)
2. Castañeda, A.G., Brown, B.J., Rusinkiewicz, S., Funkhouser, T.A., Weyrich, T.: Global consistency in the automatic assembly of fragmented artefacts. In: The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, pp. 73–80 (2011)
3. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartogr. Int. J. Geogr. Inf. Geovisualization **10**, 112–122 (1973)
4. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. IEEE Trans. Electron. Comput. **13**, 118–127 (1964)
5. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. In: Eighteenth Annual Symposium on Computational Geometry, pp. 82–87 (2002)
6. Hoff, D.J., Olver, P.J.: Automatic solution of jigsaw puzzles. J. Math. Imaging Vis. **49**, 234–250 (2014)
7. Justino, E., Oliveira, L.S., Freitas, C.: Reconstructing shredded documents through feature matching. Forensic Sci. Int. **160**, 140–147 (2006)
8. Kong, W., Kimia, B.B.: On solving 2D and 3D puzzles using curve matching. In: 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 583–590 (2001)
9. Liu, H., Cao, S., Yan, S.: Automated assembly of shredded pieces from multiple photos. IEEE Trans. Multimedia **13**, 1154–1162 (2011)
10. Radack, G.M., Badler, N.I.: Jigsaw puzzle matching using a boundary-centered polar encoding. Comput. Graph. Image Process. **19**, 1–17 (1982)
11. Richter, F., Ries, C.X., Cebron, N., Lienhart, R.: Learning to reassemble shredded documents. IEEE Trans. Multimedia **15**, 582–593 (2013)
12. Richter, F., Ries, C.X., Romberg, S., Lienhart, R.: Partial contour matching for document pieces with content-based prior. In: 2014 IEEE International Conference on Multimedia & Expo, pp. 1–6 (2014)
13. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Third International Conference on 3-D Digital Imaging and Modeling, pp. 145–152 (2001)
14. Sağiroğlu, M.Ş., Erçil, A.: A texture based matching approach for automated assembly of puzzles. In: The 18th International Conference on Pattern Recognition, vol. 3, pp. 1036–1041 (2006)

15. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147**, 195–197 (1981)
16. Stieber, A., Schneider, J., Nickolay, B., Krüger, J.: A contour matching algorithm to reconstruct ruptured documents. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) DAGM 2010. LNCS, vol. 6376, pp. 121–130. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15986-2_13
17. Tsamoura, E., Pitas, I.: Automatic color based reassembly of fragmented images and paintings. IEEE Trans. Image Process. **19**, 680–690 (2010)
18. Zhang, K., Li, X.: A graph-based optimization algorithm for fragmented image reassembly. Graph. Models **76**, 484–495 (2014)
19. Zhu, L., Zhou, Z., Hu, D.: Globally consistent reconstruction of ripped-up documents. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 1–13 (2008)