

Exploring Character Shapes for Unsupervised Reconstruction of Strip-Shredded Text Documents

Thiago M. Paixão^{ID}, Maria C. S. Boeres, Cinthia O. A. Freitas, and Thiago Oliveira-Santos

Abstract—Digital reconstruction of mechanically shredded documents has received increasing attention in the last years mainly for historical and forensics needs. Computational methods to solve this problem are highly desirable in order to mitigate the time-consuming human effort and to preserve document integrity. The reconstruction of strips-shredded documents is accomplished by horizontally splicing pieces so that the arising sequence (solution) is as similar as the original document. In this context, a central issue is the quantification of the fitting between the pieces (strips), which generally involves stating a function that associates a pair of strips to a real value indicating the fitting quality. This problem is also more challenging for text documents, such as business letters or legal documents, since they depict poor color information. The system proposed here addresses this issue by exploring character shapes as visual features for compatibility computation. Experiments conducted with real mechanically shredded documents showed that our approach outperformed in accuracy other popular techniques in the literature considering documents with (almost) only textual content.

Index Terms—Document reconstruction, shape-matching, compatibility function, Optical character recognition (OCR), modified Hausdorff distance (MHD).

I. INTRODUCTION

EVERY day, a huge amount of paper documents is destroyed manually or mechanically by paper shredder machines. Intentional shredding is highly used by companies to protect data privacy, but is also frequently associated with the illicit practice of destroying criminal evidence. To recover the lost content, forensic examiners typically try to reassemble the original document by manually arranging the paper fragments just as in a jigsaw puzzle [1]. It means those shapes, colors, content type (images, text, table, graphics, numbers, letters, and others) are used by the experts in a pre-classification stage. An important and basic clue in the manual

process is first to start the reconstruction with fragments belonging to the boundaries of the document.

The manual process, besides being slow, can also be destructive since the fragments handling can cause more material damage (alteration of the physical and chemical properties of the documents) or the loss of information (fingerprints contained on the documents). In general, the slow progress in the reconstruction is related to the following factors: i) the complexity of the document(s) to be reconstructed; ii) the mutilation process suffered by the document; iii) the quantity and shape of fragments. All these factors influence the time of reconstruction, even when performed (or aided) by expert computer systems. The quantity and shape of the fragments, however, are derived from the use of hands, scissors or shredding machines, resulting in nearly regular fragments (shredding machines) or irregular (tearing). The quantity also depends on the type of shredding machine or the number of times the person has torn the document. Therefore, the identification and cataloging of adjacent fragments, whether by manual reconstruction or by computer, is the greatest challenge to reassemble a damaged document.

Computational reconstruction has emerged in the past decade mainly motivated by historical and forensics needs [2], [3]. The laborious manual effort is alleviated by algorithms capable of assessing the fitting (compatibility) of the fragments and grouping them together optimizing the overall compatibility. Therefore, fragments are manipulated only during the preliminary acquisition procedure, and the human participation is restricted to specific interventions (semi-automatic reconstruction [3]–[5]), or even not required at all (automatic reconstruction).

This paper addresses the automatic reconstruction of text documents fragmented by modern strip-cut (i.e., only vertical) paper shredders. Unlike manual tearing or shredding with old machines [3], [6], modern shredders produce fragments with more regular shape (also called strips or shreds), and therefore reconstruction is guided basically by appearance clues. This is particularly challenging for text documents since they usually depict low color information (i.e., black-and-white appearance). Besides, the appearance of the strips boundaries is partially lost due to mechanical shredding [7], [8].

Several aspects of the reconstruction problem have been addressed for generic content documents: user interaction [3]–[5]; preprocessing by strips separation [9], skew correction [7], [10], and orientation handling [10],

Manuscript received March 19, 2018; revised August 30, 2018; accepted November 26, 2018. Date of publication December 6, 2018; date of current version March 20, 2019. This work was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil, under Grant 311504/2017-5. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Siwei Lyu. (Corresponding author: Thiago M. Paixão.)

T. M. Paixão is with the Departamento de Informática, Instituto Federal do Espírito Santo, Serra 29173-087, Brazil, and also with the Universidade Federal do Espírito Santo, Vitória 29075-910, Brazil (e-mail: paixao@gmail.com).

M. C. S. Boeres and T. Oliveira-Santos are with the Departamento de Informática, Universidade Federal do Espírito Santo, Vitória 29075-910, Brazil.

C. O. A. Freitas is with the Escola de Direito, Pontifícia Universidade Católica do Paraná, Curitiba 80215-901, Brazil.

Digital Object Identifier 10.1109/TIFS.2018.2885253

[11]; cross-cut shredding (i.e., horizontal and vertical cuts) [12]–[14]. Still, little effective progress has been verified for the challenging domain of text documents. Indeed, most of the approaches for strips compatibility evaluation relies solely on color (dis)similarities computation over boundary pixels, which implies in disregarding the damage caused by mechanical cut.

To overcome such limitations, we propose, as our main contribution, a compatibility function to quantify the fitting of the strips based on the character shapes, which are more robust and discriminative than pixel color for text documents. It assumes that the strips themselves carry on a collection of reference shapes (non-shredded characters in the inner part) which enables to fit strips side-by-side. To assess the proposed function, we integrated it to the Concorde TSP (Traveling Salesman Problem) solver [15] to obtain a complete reconstruction system. In addition, we explore the non-deterministic nature of our system (better discussed in Section III) to produce a pool of potentially distinct reconstructions, and then pick the most readable one with help of the Tesseract OCR (Optical Character Recognition) software.

Computational experiments were performed with artificially- (i.e., simulated) and mechanically-shredded documents to assess the accuracy of the proposed system against popular alternatives in literature. As our main interest is in the challenging compatibility computation, the scope of this investigation was limited to strip-cut documents, since cross-cut shredding requires a stronger focus on the optimization process. Additionally, we assume: i) correctly-oriented strips; ii) single-sided strips; iii) single-page documents. Results showed the proposed approach outperforms the state-of-the-art methods for mechanically-shredded documents (i.e., the real-world scenario) with textual content only.

Briefly, the main contributions of this article are:

- A novel unsupervised method for reconstruction of shredded documents that sets a new standard for state-of-the-art;
- A thorough performance comparison of compatibility functions, as well as of full reconstruction systems available in literature;
- A new dataset of real-shredded (i.e., mechanically) documents available to the scientific community.

The remainder of the text is organized as follows. The next section presents the related work. Section III describes the proposed reconstruction system. The experimental methodology and the obtained results are, respectively, in Sections IV and V. Finally, conclusions and future works are drawn in Section VI.

II. RELATED WORK

Ukovich *et al.* [2] claim to be the first to attempt to digitally reconstruct shredded documents. Their work, published in 2004, aimed roughly at separating strips belonging to the same document based on MPEG-7 features. Several aspects of the problem have been addressed since then, mainly dealing with image processing techniques for feature extraction, metrics for compatibilities computation, and optimization algorithms to properly arrange the strips. This review

focuses on the different approaches to compute the compatibility between strips, including related topics such as image representation and feature space.

Broadly, the reconstruction approaches can be classified as general-purpose or task-oriented, meaning they were intended for the specific domain of text documents. The first group typically computes traditional (dis)similarity measurements over boundary pixels of every two strips. De Smet *et al.* [16] describe a full reconstruction system in which compatibilities result from Euclidean distance over square blocks of RGB pixels. However, no quantitative experimental results are provided to validate their proposal. On the other hand, Skeoch [7] provides quantitative and qualitative results for a local dataset (15 documents) using several metrics (e.g., Euclidean, Manhattan, Cosine, Canberra) and the very edge pixels in both RGB and HSV color models. No significant difference in performance was verified for the many metric/model combinations. On the contrary, Marques and Freitas [8] concluded that Euclidean distance performs better than RGB for colorful documents in their own dataset (60 documents). Both [7] and [8] mention that the poorest results were obtained for text-only documents in contrast to very colorful instances (e.g., folders and magazines). Based on the observation that color is not significant for text documents, Euclidean distance came to be associated to gray-level edge pixels, as can be seen in [17]–[19].

In a more recent work, Andaló *et al.* [20] address document reconstruction as an additional application of their general square jigsaw puzzle. They adapted the metric proposed in [21], which uses the $(L_p)^q$ norm over pixel differences to quantify similarity. As features, they use the left/right boundary of a candidate tile (i.e., a strip, in the case of documents) and the content predicted using the last/first two pixels in a row of the reference tile. Prediction is based on the approximated first-order Taylor's expansion around the boundary zone. Promising results were obtained for artificial shredding using the YIQ color space, however no experiments with real-shredded data was conducted.

Unlike the aforementioned approaches, task-oriented methods explore particular features of text documents (i.e., almost black-and-white/binary appearance, text layout, symbol fitting) to design customized compatibility functions. Balme [22], whose method is employed in several works [11], [14], [23]–[28], uses the absolute-value norm over weighted black-and-white pixel differences around a reference pixel. The neighborhood (two up and two down) accounts for slight vertical displacement of text lines of paired shreds. Morandell [29] deals with this issue by calculating compatibilities based on the offset of black (text line) pixels. While these two approaches use only the boundary content, Ranca [10] leverages the inner pixels the document to be reconstructed itself to build a probabilistic model of the border pixels. Given a square neighbor around a reference pixel, the model predicts the corresponding boundary pixel of the adjacent shred. Only simulated shredding with and without artificial global noise was considered in experiments.

Gong *et al.* [27] observed the discriminative power of the Balme's function [22] is diluted due to ambiguities caused by

background-background (white-to-white) accumulated comparison. Therefore, they additionally quantify the coherence of empty (background) rows of pixels, which implicitly penalizes vertical misalignment of text lines. Alternatively, text lines can be explicitly detected for further quantification of vertical displacement [13], [30]. Although this is less sensitive to the damage on strips boundaries, well-structured text documents still cause ambiguities in the strips compatibility verification. Pöhler *et al.* [5] encode the boundary region of the strips at text- and paragraph-level using HSV color-based features. They concluded, however, their method demands supplementary semantic information (e.g., word-level analysis) to perform accurately without human intervention (i.e., fully automatically).

Reconstruction can also benefit from characters clues for pairing shreds. Perl *et al.* [31] investigated OCR features for supervised recognition and matching of characters. Instead of performing recognition, Phientrakul *et al.* [32] explore the linear trend of the character strokes at the sectioned edges to predict the content of adjacent strips. In a similar approach, Guo *et al.* [33] quantify strokes discontinuities based on image gradient. The main limitation in [32] and [33] is that the linearity assumption does not hold for real-shredded instances since part of the character can be lost, or its two parts (in different strips) can be vertically misaligned. Language-restricted methods (e.g., Chinese documents reconstruction [34], [35]) can also take advantage of from meaningful character structural features of a particular alphabet. Xing *et al.* [34] use only the horizontal strokes of Chinese characters for fitting purposes. Xing and Zhang [35] propose a probabilistic model for combinations of character fragments. The model is built in a self-supervised approach by collecting statistics of character structural properties from adjacent strips of a training set of documents.

In summary, it has been noticed a growing interest in the document reconstruction topic over the last years. Nonetheless, mainly from a computer vision perspective, the compatibility assessment between strips is still an open issue for text documents. In this context, boundary pixels alone are not enough to achieve good reconstructions, and this fact can not be verified with experimentation restricted to artificially-shredded data, as occurs in [11], [13], [20], [27]–[29], [32], and [33]. The proposed character shape-based approach, described in the next section, was designed considering a more realistic scenario where strips edges may be significantly corrupted.

III. PROPOSED SYSTEM

The proposed reconstruction system (Figure 1) takes as input a set document strips (digital images), and outputs, as final solution, the reconstruction of the strip-cut document (i.e., strips permutation). The full system is divided into four main stages and its basic workflow is as follows.

Initially, text regions are coarsely extracted from each strip, and characters in each region are segmented and partitioned into inner and edge characters. These sets are kept for further processing, however strips without inner characters are

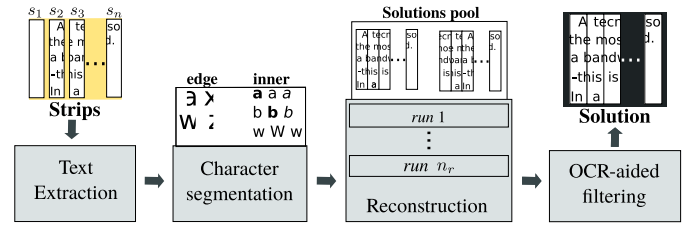


Fig. 1. Overview of the proposed reconstruction system for strip-cut documents. Firstly, text regions are roughly extracted from the individual input strips. Thereafter, the system segments (and separates) the inner and edge characters inside each text region. These characters feed the reconstruction algorithm, which produces a pool of n_{sol} candidate reconstructions. Finally, the OCR-aided filter elects the most readable reconstruction in the pool.

assumed as blank, and they are disregarded in the final reconstructed document. At this point, it is important to mention that the reconstruction takes place by verifying the fitting of edge characters between every pair of strips. For this purpose, a subset of inner characters (representative characters) is used as reference for shape matching. In a broader view, the reconstruction stage uses the characters information to produce a pool of n_{sol} candidate solutions, one for each run of the non-deterministic character shape-based algorithm (discussed later in Section III-C). In the last stage, the OCR-aided filter selects the most readable solution from the pool potential candidates, i.e., the solution with the maximum number of recognizable words by a third-party OCR software. These stages are detailed in the following sections.

A. Text Extraction

Documents with textual content may also contain figures, tables and other visual elements that can hamper the character segmentation accuracy. To avoid this, strips are preprocessed in order to differentiate text regions from graphic components. The full procedure, described in the ensuing paragraphs, assumes a set of constraints (empirically adapted from [36]) on text properties defined over document scanning resolution in pixels/mm (R): $H_{tmin} = 1.8R$, $H_{tmax} = 5.5R$, and $D_{cmax} = 1.2R$, where P_{min}/P_{max} denotes the minimum/maximum allowable value for a property P ; H_t and D_c denote, respectively, the properties text line height and same-line characters distance. Small variations on these parameters do not have significant influence on the system final accuracy, as assessed in the experimental results (Section V).

To extract text, each strip image is firstly globally thresholded by applying Otsu's algorithm [37] to separate potential text regions (objects) from background. Text lines structure arises by merging close objects in the horizontal direction, so that characters belonging to the same line can be grouped together. This is achieved by carrying out the morphological dilation on the thresholded image using a $d_x \times d_y$ 8-connected structuring element, where d_x is set to $2D_{cmax}$, and d_y is set to $d_x/4$. The structuring element shape reflects the aspect ratio of a text line and the direction of splicing.

In this stage, small artifacts (including noise) and remaining graphical objects included among the text region candidates may be observed. To keep only text, a filtering scheme



Fig. 2. Overview of the character segmentation procedure. Red boxes delimit inner characters, while blue and green boxes delimit, respectively, the left and right edge characters.

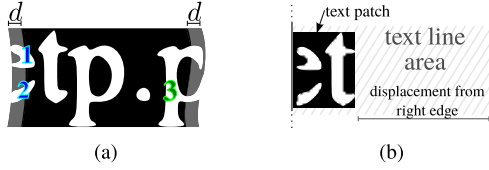


Fig. 3. Edge characters segmentation considering d -thick boundary zones. (a) Left edge character is the union of components 1 and 2, as the right edge character comprises only the component 3; (b) The text patch does not span the entire text line.

is applied over the connected components based on their bounding boxes dimensions, as well as on the previously defined constraints. Let $B = (x_B, y_B, w_B, h_B)$ be one of these bounding boxes located at (x_B, y_B) , with dimensions $w_B \times h_B$. Two situations are considered here:

- 1) $h_B < H_{min}$: in this case, the connected component associated with B is simply removed from the candidates.
- 2) $h_B > H_{max}$: for this situation, not only the component B is removed, but also the components which intersect the space vertically delimited by y_B and $y_B + h_B - 1$.

B. Character Segmentation

In this stage, the characters contained in the rectangular patches are segmented, as depicted in Figure 2. First, an initial segmentation is performed to establish the bounding boxes of both: the left edge (in blue) and right edge (in green) characters in a patch. Subsequently, the system segments the inner characters (in red), i.e., those which are horizontally enclosed in the area between the edge boxes. Outliers patterns of these inner characters are removed in the last stage. The following subsections describe these steps in greater details.

1) *Edge Characters Segmentation*: An edge character encompasses one or more connected components which are close to some extremity of its respective strip, as seen in Figure 3a. Closeness is determined with respect to a boundary zone that extends d pixels from the left/right border inside the strip. Then, the union of components that intersect the left/right boundary zone is taken as a single edge character. Note in this illustration that the depicted text patch occupies a full text line, and the left edge character comprises the components 1 and 2, while the right includes only the component 3. The d displacement value was empirically adjusted to $0.5R$, where R is the document scanning resolution introduced in Section III-A).

As depicted in Figure 3, edge characters are potentially cut, although they also may be entirely preserved (complete character), which happens when the cut goes along the inter-character space. In addition to this observation, it is worth to mention that text patches not always match the full text line

area. In this case, the rightmost character in a patch may not be an edge character, as illustrated in Figure 3b.

2) *Inner Characters Segmentation*: Inner characters may be composed of two or more connected components, depending on the quantity of noise in the scanned document, as well as on the presence of punctuation and accent symbols. Segmenting these characters is accomplished by simply grouping components, and a post-processing filtering step.

In the grouping stage, such components are partitioned so that each subset \mathcal{C} is a maximal set satisfying the following property: $\forall c \in \mathcal{C}, \exists c' \in \mathcal{C}, c$ intersects vertically c' . Filtering consists in keeping, for each subset, only that component with the largest bounding box. Thus, what we call inner character is, in fact, the main shape compounding the character.

3) *Removal of Inner Characters Outliers*: As discussed so far, the proposed system takes advantage of little prior information to extract text and segment characters from strips. A few conservative set of constraints was established in order to enable the reconstruction of documents in a variety of font sizes. As a consequence, outlier patterns caused by broken and merged characters may be misassigned as inner characters after segmentation even with filtering scheme.

We observed large outliers are less frequent given their inherent dimensions, and thus can be naturally disregarded with the extraction of the representative inner characters (further discussed in Section III-C.1). Hence, this stage addresses only the removal of small outliers (with respect to their height and bounding-box area) in order to achieve a more reliable dataset. This is achieved in the following manner:

- 1) Compute, for all inner characters, the medians of the bounding box height and area medians, H_{med} and A_{med} , respectively;
- 2) Exclude inner characters whose bounding box height is less than $0.4H_{med}$, or bounding box area is less than $0.6A_{med}$.

It is interesting to observe that this operation not only removes unlikely symbols, but also recognizable characters whose dimensions differ strongly from the main text content, which are unhelpful to determine the fitting between strips.

C. Character Shape-Based Reconstruction Algorithm

The reconstruction algorithm is the core of the proposed system. It is designed to provide a solution given i) the full set of inner characters, and ii) all the strips with some text content, as well as their respective edge characters. Figure 4 provides an overview of a single run of the algorithm, which is briefly described in the next paragraph.

Firstly, similar-shape inner characters are clustered together so that only a single representative character per cluster is kept for further computations. The full set of representative characters constitutes all the system knowledge about character shapes, and is used to evaluate the pairwise compatibilities. These values are computed by joining strips horizontally and evaluating how the emerging character patterns fit each other. Such patterns arise near to the touching edge, where the rightmost characters of the left strip merge to the leftmost characters of the right strip. The resulting compatibility values

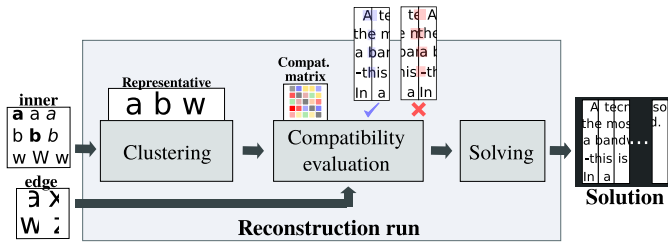


Fig. 4. A single run of the algorithm produces a candidate solution based on the previously segmented inner and edge characters. Firstly, a set of representative is extracted from the inner characters as a byproduct of the clustering algorithm. Next, the pairwise compatibilities are computed by verifying the adequacy between the emerging character patterns (when two strips are spliced together) and the representative characters. A solving procedure is conducted in the end to produce a candidate solution.

are arranged in a square asymmetric matrix, which is the input for the solving stage. Our approach yields the optimal (maximum compatibility) solution for an instance by reformulating the original reconstruction problem as a Traveling Salesman Problem (TSP) and then taking advantage of Concorde [15] TSP solver. Optimality is achieved when Concorde is used in conjunction with the QSOpt linear programming solver [38]. The steps of the reconstruction algorithm are described more detailed in the next subsections.

1) *Clustering*: The inner characters are clustered by applying an efficient k -medoids algorithm called CLARANS [39]. The algorithm operates over a pairwise dissimilarity matrix resulting from applying the Modified Hausdorff Distance (MHD) [40] metric on every two (inner) character shapes. For more accurate calculation, the shapes are aligned with respect to their centroids before the MHD computation, which, in turn, takes into consideration only the shapes' contours for speed-up purposes [41].

The number of clusters (k in k -medoids) was set to 52 to accommodate the 26 letters of Latin alphabet in both lower and uppercase modes. The other parameters values, i.e., the number of local searches and the maximum number of neighbors, were set as recommended by Ng and Han [39], which are, respectively, 2 and $0.0125 \times n_{objs}(n_{objs} - k)$. The n_{objs} denotes the number of objects (inner characters) to be clustered.

2) *Compatibility Function*: A central question in the reconstruction problem is to determine whether two arbitrary distinct strips s_i and s_j were neighbors in the original document. Since this true-or-false question is hard to be algorithmically answered, the most common approach consists in quantifying the fitting of s_j when placed right next to s_i . This is done by defining a compatibility function $\gamma : S^2 \rightarrow \mathbb{R}$, where $S = \{s_1, s_2, \dots, s_m\}$ denotes from this point on the set of the m non-blank strips remaining at this stage. The pairwise compatibilities computed over S can be arranged in a $m \times m$ matrix C , where each entry corresponds to

$$C_{i,j} = \begin{cases} \gamma(s_i, s_j), & \text{if } i \neq j \quad i, j = 1, 2, \dots, n \\ +\infty, & \text{otherwise.} \end{cases} \quad (1)$$

Typically, it is observed that, for $i \neq j$, $\gamma(s_i, s_j) \neq \gamma(s_j, s_i)$, which stresses the asymmetric aspect of the compatibility matrix. Throughout the rest of this section, $\gamma(s_i, s_j)$ denotes the numerical value resulting from applying the

Algorithm 1 Classifying the Character Type: Empty (E), Complete (C), or Fragmented (F)

```

1: procedure CHAR-TYPE(char)
2:   if NULL(char) then
3:     return "E" ▷ Empty
4:   else if MHD(char, rep) <  $\tau_{shape}$ , for some representa-
     tive character rep then
5:     return "C" ▷ Complete
6:   else return "F" ▷ Fragmented
7:   end if
8: end procedure

```

compatibility function over the strips s_i and s_j . The compatibility value for a pair of strips, as illustrated in Figure 4, is derived from the characters associations (blue and red boxes) emerging around the touching edge.

Since the compatibility function explores the character shapes to score the associations, a shape matching criteria has to be defined. In this work, two shapes A and B match iff $MHD(A, B) < \tau_{shape}$, where τ_{shape} is a threshold determined experimentally, as discussed in Section IV-C. Scoring an association depends essentially on classifying the type of its elements (Algorithm 1), which can be an edge character (fragmented or complete), or even the absence of information (E). An edge character is considered complete (C) if it matches a representative character, otherwise it is fragmented (F). Based on this terminology, the associations can be divided into five general types:

- 1) *Fragment-Fragment* (FF): both the characters are fragmented.
- 2) *Character-Character* (CC): both the characters are complete.
- 3) *Empty-Fragment* (EF): there is a single character and it is fragmented.
- 4) *Empty-Character* (EC): there is a single character and it is complete.
- 5) *Fragment-Character* (FC): one character is fragmented, and the other is complete.

The terms EF, EC, and FC do not denote the elements order, i.e., an association FC can stand for a fragment followed by a complete character or vice versa. FF is further subdivided into two categories: *Matching Fragment-Fragment* (FFm), where two merged fragments matches a representative character, and the complementary *Non-matching Fragment-Fragment* (FFn).

Table I summarizes all types of associations considered in this work relating them to positive (+1), neutral (0), or negative ($-p$) scores. The positive score, assigned to FFm and CC, contributes to the belief that the respective strips are adjacent. Oppositely, the associations EF, FFn, and FC reflect undesirable situations, and then a penalty score (with $p = 0.2$) is assigned to them. EC is assumed to be neutral, i.e., zero score. Some associations may be visually ambiguous, even for human readers. For instance, two letters "v" (i.e., "vv") may be interpreted as CC, or as FFm in view of the resemblance to the "w" shape. To tackle this, we assume a priority for

TABLE I
ASSOCIATION TYPES AND THEIR RESPECTIVE SCORES

Association type		# Entries	Score
Matching Fragment-Fragment (FFm)		2	+1
Character-Character (CC)		2	
Empty-Character (EC)		1	0
Empty-Fragment (EF)		1	
Non-matching Fragment-Fragment (FFn)		2	-p
Fragment-Character (FC)		2	

Algorithm 2 Scoring an Association of Edge Characters

```

1: procedure SCORE(char1, char2)
2:   type1 ← CHAR-TYPE(char1)
3:   type2 ← CHAR-TYPE(char2)
4:   if type1 = “E” then
5:     ▷ EC or EF
6:     if type2 = “C” then return 0 else return -p
7:   else if type2 = “E” then
8:     ▷ EC or EF
9:     if type1 = “C” then return 0 else return -p
10:  else
11:    charm ← MERGE(char1, char2)
12:    if MHD(charm, rep) < τshape, for some representa-
        tive character rep then
13:      return 1                                ▷ FFm
14:    else if type1 = “C” and type2 = “C” then
15:      return 1                                ▷ CC
16:    else
17:      return -p                                ▷ Remaining (FFn, FC)
18:    end if
19:  end if
20: end procedure

```

FFm for associations without empty characters, as detailed in Algorithm 2.

The compatibility of a pair (s_i, s_j) , $i \neq j$, is then the summation of the individual scores for each association $(char_i, char_j)$. Let $A_{i,j}$ be the set of associations for (s_i, s_j) . Then, based on the Algorithm 2, the compatibility is formally defined as

$$\gamma(s_i, s_j) = \sum_{(char_i, char_j) \in A_{i,j}} \text{SCORE}(char_i, char_j). \quad (2)$$

3) *Solving*: The final stage aims to find an optimal strips arrangement (permutation) according to the pairwise compatibility matrix \mathbf{C} obtained in the previously step. As we leverage a TSP-based solver, compatibilities should be first converted to (non-negative) distances arranged in a matrix \mathbf{D} . In this paper, we adopt $\mathbf{D} = \max(\mathbf{C} - \text{diag}(\mathbf{C})) - \mathbf{C}$, where $\max(\mathbf{M})$ denotes a matrix with the same dimensions as \mathbf{M} whose elements are equal to the maximum value of \mathbf{M} , and $\text{diag}(\cdot)$ denotes the diagonal matrix of a matrix. Then, the aimed solution is

a permutation $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ of $\{1, 2, \dots, m\}$ which minimizes the summed costs (i.e., distances) between adjacent strips in the reconstruction induced by π , or more formally,

$$\phi(\pi) = \min \sum_{i=1}^{m-1} \mathbf{D}_{\pi_i, \pi_{i+1}}. \quad (3)$$

The matrix \mathbf{D} can be viewed as a directed weighted graph $G = (V, A, w)$, where vertices in V are uniquely associated to strips, and $w(a)$, given an arc $a = (v_i, v_j) \in A$, carries the $\mathbf{D}_{i,j}$ value as weight. A solution $v_{\pi_1} v_{\pi_2} \dots v_{\pi_n}$ for Minimum-Cost Hamiltonian Path Problem (MCHPP) on G induces a permutation $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ which is the searched solution for the reconstruction problem. This combinatorial formulation is inspired in the work of Prandtstetter and Raidl [11] in which the reconstruction problem is reformulated as a TSP. In our work, MCHPP is not solved directly since the Concorde solves the symmetric TSP. Our strategy is to reduce MCHPP to the Asymmetric Traveling Salesman Problem (ATSP) by adding a dummy vertex v' into the original graph, and also adding a set of zero-weight arcs, A_0 , connecting v' to the previously existing vertices. More formally, an ATSP instance $G' = (V', A', w')$ arising from $G = (V, A, w)$ is such that $V' = V \cup \{v'\}$, $A' = A \cup A_0$, where $A_0 = \bigcup_{v \in V} \{(v', v), (v, v')\}$, and, for all $a' \in A'$,

$$w'(a') = \begin{cases} w(a'), & \text{if } a' \notin A_0 \\ 0, & \text{otherwise.} \end{cases}$$

Let the cycle $C = v_{\pi_1} v_{\pi_2} v_{\pi_3} \dots v_{\pi_n}$ be a solution for ATSP, and assume v_{π_1} is the aforementioned dummy vertex. A solution for MCHPP is obtained by removing v_{π_1} and its incident arcs, which results in the simple path $v_{\pi_2} v_{\pi_3} \dots v_{\pi_n}$. ATSP is indirectly solved by reformulating it as a TSP [42], and then invoking Concorde.

D. OCR-Aided Filtering

The reconstruction stage returns n_{sol} solutions by running multiple times the non-deterministic algorithm described in Section III-C. This strategy avoids placing trust in a single solution, however it demands a selection criteria to point the final solution. Our OCR-aided filter outputs the most readable reconstruction as the final solution, i.e., that with the highest number of recognized words (minimum of three characters) according to an input dictionary. Such words are counted by following three steps:

- 1) Text recognition: arrange the strip images side-by-side given the order established in a solution (permutation), and then run an OCR Engine (Tesseract [43]) on the whole image;
- 2) Tokenization: remove any punctuation and tokens (i.e., sequences of less than three symbols separated by blank spaces);
- 3) Word count: check how many tokens are found in the dictionary.

IV. EXPERIMENTAL METHODOLOGY

The experiments aim primarily to evaluate the proposed system in terms of solutions accuracy, as well as to provide a comparative evaluation against relevant reconstruction approaches in literature focusing on the effectiveness of compatibility functions. The influence of real-shredding against the artificial process is also investigated in this work. The following sections discuss the datasets used in the experiments, the adopted accuracy metric, and, ultimately, the details of the experimental procedure, including the hardware/software specification.

A. Datasets

The two test datasets¹ used in the experiments, referred as D1 and D2, consist of digital strips obtained from a set of reference documents following two distinct processes:

- Artificial (art): algorithmic process that virtually cuts the digital document in 30 equal-width pieces.
- Mechanical (mec): real process which requires printing out the reference documents, submitting the printed documents to a shredder machine, and finally scanning the shreds at 300 dpi (≈ 11.81 pixels/mm).

The dataset D1 is composed of D1-mec, a set of 60 mechanically-shredded text documents provided by Marques and Freitas [8], and D1-art, the respective artificially-cut documents. D1-mec was made available in separated and scanned strips, while the D1-art was generated by us based on the reference scanned documents also provided with the strips dataset. The D1 documents were visually classified in three categories for further analysis: 39 text-only documents (TO), 9 documents with line-based graphics (LG) – which includes diagrams and tables –, and 12 documents containing filled-graphics (FG), such as photos and colorful images. To be categorized as LG or FG, a document must have a considerable portion of its inner area occupied with graphical elements, which implies that documents with small or peripheral graphics, such as company logos, are categorized as TO.

The dataset D2, in turn, is another contribution of this work. It was assembled based on 20 text-only scanned documents (business letters and legal documents) from the ISRI-Tk OCR dataset [45], being D2-mec generated by using a Leadership 7348 paper shredder. The dataset D2-mec was created by scanning strips against a paper sheet with a high contrast color, which facilitated the further semi-automatic segmentation (see Figure 5). For this purpose, we assume that every document pixel belongs to one of the following three classes: i) strip object, which includes dark pixels of text regions and graphical elements; ii) strip paper, that is, the background of a strip; iii) paper substrate, the colorful paper portion used to support the strips. Based on these assumptions, strips segmentation can be achieved by identifying and removing the paper substrate area, which is done in two steps. First, we employ the classical

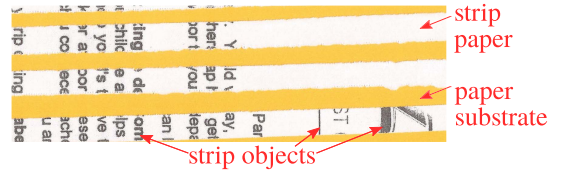


Fig. 5. Strips belonging to the same document are arranged together onto a paper sheet with high contrast (D2-mec).

k -means algorithm to cluster the image pixels in three groups according their RGB color. In a second moment, the label corresponding to the paper substrate is manually identified by an user, so that every pixel with the same label can be removed.

Comparatively, documents in D1 are more heterogeneous than those in D2 with respect to text layout structure, which led us to analyze them in categories. The greater the proportion of pictorial content (i.e., categories LG and FG) on the document is, the more challenging the reconstruction instance for the proposed approach is. D2 documents are, nevertheless, more noisy, mainly due to shadow effects produced during scanning. Besides, the paper shredder used to assemble D2-mec produces more curved strips than in D1-mec, as well as it causes more damage on the strips borders.

B. Accuracy Metric

This work uses a neighbor comparison metric [20], which counts the number of well-ordered subsequences (here just called groups) of a given solution [11], [29]. Formally, a group in a solution $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ is a maximal m -tuple $(\pi_k, \pi_{k+1}, \dots, \pi_{k+m-1}, \pi_{k+m})$, $m \leq n$, such that $\pi_{i+1} = \pi_i + 1$, for all $i = k, k+1, \dots, k+m-1$, assuming $(1, 2, \dots, n)$ as the correct solution. Thus, the number of groups in π can be expressed as

$$ngroups(\pi) = n - \sum_{i=1}^{n-1} \mathbf{1}\{\pi_{i+1} = \pi_i + 1\}, \quad (4)$$

where $\mathbf{1}\{\cdot\}$ is the true-or-false indicator function. Note that the correct solution has only a single n -size group, while a fully disordered solution has n one-size groups. The final accuracy is measured as

$$accuracy(\pi) = \frac{n - ngroups(\pi)}{n - 1}, \quad (5)$$

so that 0 (the minimum value) points out the worst quality solution, and 1 (the maximum value) reflects the best.

C. Experiments

In the first experiment, the proposed system was run over the datasets D1 and D2 for 10 times while computing the accuracy for every final solution, i.e, the solution selected by the OCR-aided filter (see Figure 1) from the pool of the $n_{sol} = 10$ candidate solutions. This process totals 100 solutions for each test instance (i.e., 10 candidate solutions for each of the 10 runs), with only 10 final most readable solutions. To evaluate the effectiveness of the OCR-aided filter, we measured

¹The main public datasets [6], [44] related to this work could not be used because the type of document addressed here (e.g., business letters, legal documents, and technical reports) differs significantly from such datasets.

the percentage of system runs the highest accuracy solution is selected from the pool.

The threshold for shape matching (τ_{shape}) was calibrated over a separated collection of 10 artificially-shredded documents from ISRI-Tk OCR. For a more realistic shredding simulation, it was applied additive Gaussian white noise over the three first/last pixel-columns. The Gaussian distribution was configured to zero-mean, and to standard deviations 200, 150, and 100 (each value applied over a single pixel-column ranging from the border inside). The system was run once for each $\tau_{shape} \in \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$, producing a pool of $n_{sol} = 10$ candidate solutions for each value. The chosen threshold (0.5) is that for which the system (without the OCR-filter) achieved the highest average accuracy regarding to the pool of solutions.

We also conducted an experiment to evaluate (one-factor-at-a-time) the sensitivity of the system with respect to the text extraction parameters H_{tmin} , H_{tmax} , and D_{cmax} (introduced in Section III-A). Each parameter is assigned a set of evenly spaced values ranging from 80 to 120% of its original value. The system (without the OCR-filter) was run once for each document/parameter configuration considering only documents in D1-mec and D2-mec. Thus, for each parameter configuration, a total of 800 solutions ($n_{sol} = 10$ candidate solutions for each one of the 80 documents) was generated.

For comparative analysis, we investigated the following reconstruction methods (referred by the first author's name): Sleit [13], Marques [8], Balme [22], Morandell [29], and Andaló [20]. Only Andaló provided the original system implementation. It is basically a gradient-based solver for the square jigsaw puzzle which can be also applied for the document reconstruction problem. Marques' system uses a simple nearest-neighbor search and was fully implemented from the scratch. Balme's system was proposed in a technical report which is not publicly available. Nevertheless, the compatibility function was seamlessly implemented based on the several works that make use of it [11], [14], [23]–[28]. The Sleit's custom search heuristic and the Morandell's meta-heuristic formulation could not be implemented, however their compatibility functions were assessed in the experiments.

Based on these observations, we directly compared the accuracy of our system with Andaló and Marques. Additionally, we evaluated the compatibility functions of all the methods in conjunction with our optimal solving scheme (introduced in Section III-C.3). This focuses the analysis on the main issue investigated in this paper: the influence of compatibility functions in the quality of the reconstruction. The prefix "Concorde/" was placed before the method identification (i.e., the first author's name) to refer to the modified version (e.g., Concorde/Morandell). Full system and compatibility functions evaluation followed the same experimental procedure. First, blank strips were manually removed for the compared methods since either they do not handle this issue in the original formulation, or the proposed strategies are suitable only for artificially-shredded documents. Parameters were set as recommended by the authors. For Andaló and Concorde/Andaló, however, a more conservative approach was adopted. The two sets of parameters recommended by the authors were

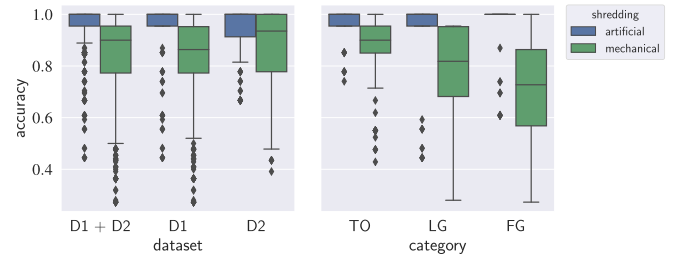


Fig. 6. Accuracy (presented as median boxplots) of the proposed system considering artificial (blue bars) and mechanical (green bars) shredding. Left: the first group shows the resulting accuracy for the full documents collection (D1 + D2), while the other two are related to each individual dataset. Right: accuracy of the proposed system across categories for the dataset D1.

tested and only the highest accuracy was reported. For all methods, 11 configurations resulting from removing 0, 1, 2, ..., or 10 pixels in each strip row were examined. Similarly, only the highest accuracy solution for each method/document instance was regarded for further analysis.

D. Experimental Platform

The experiments were conducted on two different machines. The first, an Intel® Core™ i3-2100 PC (3.10GHz) with 4GB of RAM, was used for accuracy assessment and system calibration. For time performance test and sensitivity analysis, we used an Intel® Xeon™ E7-4850 v4 (2.10GHz) with 128 vCPU (only 10 were used), 252GB. The experiments can be easily reproduced² thanks to the Docker technology [46] that manages all the environment configuration. The methods were implemented in Python with support of OpenCV for high-performance image processing, except for Andaló, whose original C++ implementation was provided by the respective authors.

V. RESULTS AND DISCUSSION

This section starts with the discussion of the experimental results for the proposed system. In the second part, we compare our method with the literature.

A. Proposed System Evaluation

The left graph of Figure 6 summarizes the resulting accuracy (Equation 5) for the solutions obtained with the proposed system. The results were arranged in three groups paired according to shredding type (artificial or mechanical). The first group depicts the overall performance regarding the full documents collection (D1 + D2). The two remaining groups cover, respectively, the datasets D1 and D2 in isolation. By comparing mechanical to artificial shredding, we confirm the decay in accuracy, being 0.136 (on average) for D1, and 0.064 for D2. This means that the solutions for mechanical shredding presented, respectively, about four and two more groups (on average) considering 30-strips size shredded documents (each new group causes a decay of ≈ 0.034 in accuracy).

²The software, dataset, and running instructions will be public available at <https://github.com/thiagopx/docrec-tifs18>.

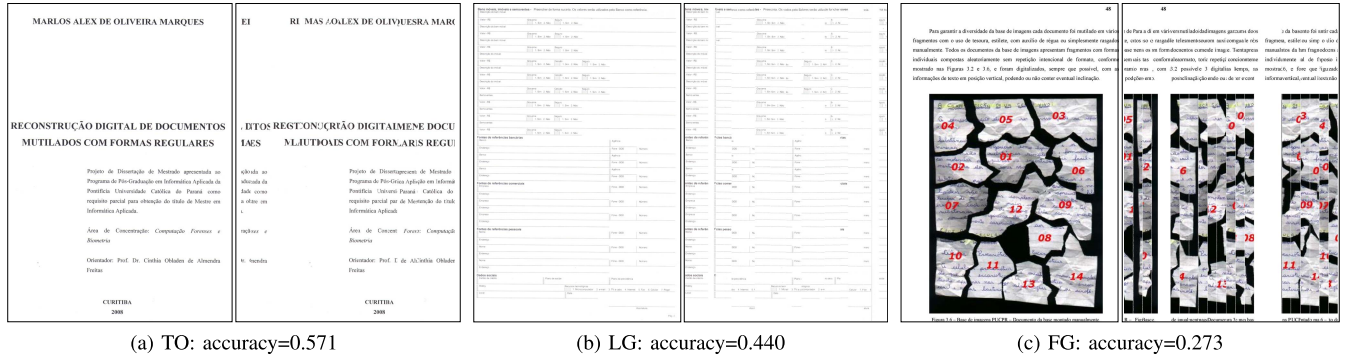


Fig. 7. Challenging test instances due to poorly structured text areas: (a) large title characters are ignored by the reconstruction algorithm; (b) form depicts sparse text with tiny characters; (c) large figures fill most of the document area.

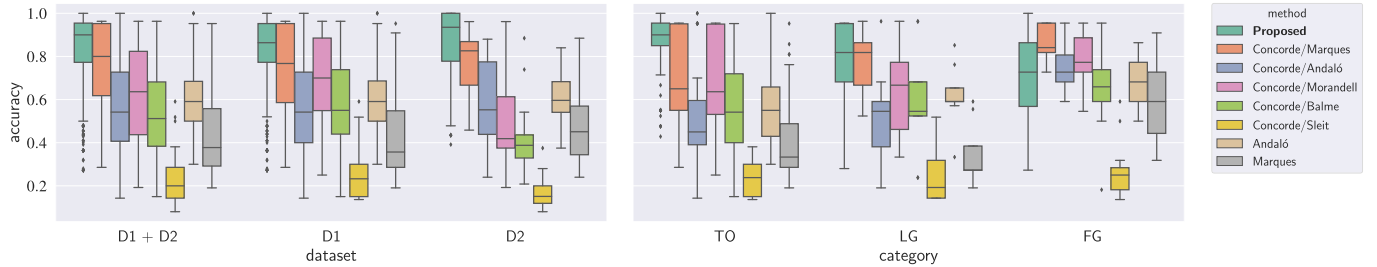


Fig. 8. Comparative results for mechanically-shredded documents. Left: resulting accuracy for D1-mec and D2-mec analyzed in conjunction and separately. Right: detailed results for D1-mec across categories.

The right graph of Figure 6 shows the accuracy for D1 with respect to the three documents categories introduced in the previous section. On average, the accuracy for mechanical shredding drops 0.088, 0.148, and 0.280 for text-only (TO), line-based graphics (LG), and filled-graphics (FG) documents, respectively. Since we measure compatibility at character level, the reconstruction quality is less sensitive to mechanical shredding when then documents have rich structured textual content, as usually occurs for the TO category. The main reason for this is that the more structured the text in a document is, the more edge characters associations available for the scoring procedure (Algorithm 2) there are. With a large sample of associations, compatibility scoring is less sensitive to the loss of matching associations due to document damage caused by mechanical shredding. On the contrary, a reduced quantity of associations makes the scoring procedure more sensitive to the loss of matches, which is more likely in LG and FG documents since they carry graphical elements in addition to the text content. Note that the presence of graphics is not a problem itself, as confirmed by the results achieved for LG and FG with artificial shredding. However, the greater the area covered by figures is, the lesser the amount of text available to produce character associations is. Figure 7 shows three challenging instances with poorly structured text.

The performance of the proposed system also counts on the OCR-aided filter ability for choosing the highest quality (accuracy) solution from the pool of candidate solutions. The filter worked properly in 86.56% of the tested cases, including all the documents (D1+D2) for both artificial and mechanical shredding. Restricting the domain for text-only documents, this value is slightly increased to 87.80%.

For sensitivity analysis, we measured the average accuracy for the different configurations of the text extraction parameters. The minimum and maximum average accuracy were, respectively, 0.797 and 0.839. In other words, individually moving the parameters up to $\pm 20\%$ from their original values causes around 4.22% of variation on the average accuracy. Considering a window of $\pm 10\%$, the accuracy variation is reduced to 2.61%.

B. Comparative Analysis

The left graph in Figure 8 shows the comparative results (only mechanical shredding) including two original systems (Andaló and Marques) and five modified systems, i.e., the original compatibility functions coupled to our solving scheme. On average, our system achieved the best overall performance (0.843), followed by Concorde/Marques (0.743), while Sleit presented the poorest accuracy (0.226). Note that Concorde/Andaló outperforms Concorde/Morandell and Concorde/Balme only for D2. The compatibility functions of the latter two methods were intended to deal with ideal black-and-white documents, and have been shown very sensitive to noisy borders.

The compatibility function of Marques performs better when coupled to the optimal Concorde solver (Concorde/Marques) than to the original nearest-neighbor algorithm. Although this was expected, given the optimality of the Concorde solver, the opposite behavior was verified for Andaló. As presented by Andaló *et al.* [20], the compatibility function parameters were calibrated specifically to optimize the performance of their gradient-based solver. Nevertheless, the range of the float compatibilities produced with the

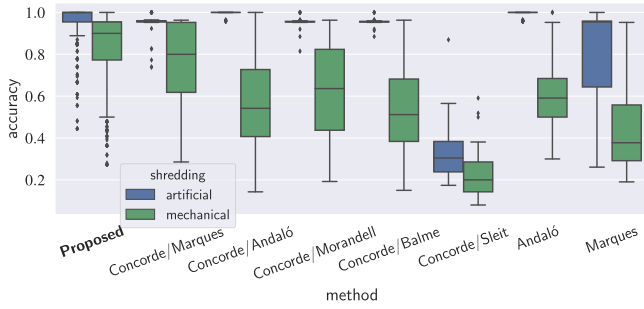


Fig. 9. Comparative results (overall accuracies) for both artificially- and mechanically-shredded documents.

calibrated values (i.e., the recommended parameters) tends to be more compressed than for other methods. Therefore, the Concorde/Andaló accuracy is degraded due to loss of precision caused by successive numerical operations to transform compatibilities into valid integer inputs for Concorde.

The most significant results obtained with our system concern, as expected, to the reconstruction of text-only documents. This can be noticed on the left graph of Figure 8, where our system achieved the highest average accuracy of 0.872 for D2, and also in the right graph of Figure 8, where the average accuracy was 0.888 for TO documents in D1. On the contrary, the other methods perform better when the documents are rich in graphics (LG and FG), as seen in the right graph of Figure 8.

The drop in accuracy for the compared systems/functions is also remarkable when tested with real-shredded documents. This is reflected in Figure 9, which depicts the overall accuracy for both datasets. With exception of Concorde/Balme, Concorde/Sleit, and Marques, the compared methods achieved accuracy superior to 0.950 with low variability when tested on artificially-shredded documents. In contrast, the green bars show the marked performance degeneration when real-shredding is addressed. This stresses the fact that the experimentation with artificially-shredded documents alone is inconclusive for the reconstruction application.

The accuracy performance of the proposed system, however, comes with a time cost that increases with the number of associations to be evaluated. For instance, the average time (in 10 runs) our system took to reconstruct a document with high text density (1308 characters) was 372.22s (6m12s), while for Concorde/Marques, the second-best performed, the time was 3.87s. To alleviate this problem, some issues should be addressed: i) parallel computation of the pool of candidate solutions, since they are independent of each other (a preliminary test yielded 2m39s for the same reconstruction instance); ii) full implementation of the system in a compiled language (e.g., C/C++); iii) extensive investigation of cheaper metrics for shape matching than the Modified Hausdorff distance.

VI. CONCLUSIONS AND FUTURE WORK

This paper addressed the problem of reconstruction of strip-shredded text documents by proposing a system to achieve this end. The success in the reconstruction task depends strongly on the compatibility function to verify matching strips. While most of the designed functions in literature has made use of

strip border pixels as visual features, the proposed approach exploits one level up of abstraction by matching character shapes instead of pixels. The proposed method assumes that all the necessary character shape information is contained in the documents themselves.

The experiments pointed that our approach achieved the best results for documents with rich text content, which was expected since character information is the relevant feature used to match strips. In this context, that includes the dataset D2 and the TO documents in D1, the average accuracies (mechanical-shredding) for the proposed system were, respectively, 0.872 and 0.888, while the second best method (Concorde/Marques) attained 0.782 and 0.679, even considering only the top accuracy solution for each document.

Despite of promising results, the time performance of the proposed system is still an open issue. Future work should investigate the impact of faster computing metrics for character shape matching on the accuracy of our system. Besides, strategies to reconstruct documents based on partially computed compatibility matrices should be studied. This can be done by detecting unlikely pairing of strips, and assigning to them a low compatibility score.

The multi-documents reconstruction is also another direction of our future work. To tackle this, the compatibility function must be adapted to become font-size independent. A possible solution for such question is the application of size normalization strategies on the characters, although this is not trivial in the case of fragmented characters. In addition, dealing with mixed strips in a more real scenario necessarily assumes the presence of graphics. Exploring this content properly is crucial to produce better solutions, and may be achieved by blending our compatibility function with those based on border pixels.

Finally, the use of deep learning in different levels of our system should be investigated. As the start point, it would be interesting to evaluate convolutional networks for end-to-end text extraction instead of the current parametric algorithm.

REFERENCES

- [1] D. J. Hoff and P. J. Olver, "Automatic solution of jigsaw puzzles," *J. Math. Imag. Vis.*, vol. 49, no. 1, pp. 234–250, 2014.
- [2] A. Ukovich, G. Ramponi, H. Doulaverakis, Y. Kompatsiaris, and M. G. Strintzis, "Shredded document reconstruction using MPEG-7 standard descriptors," in *Proc. 4th IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2004, pp. 334–337.
- [3] P. Butler, P. Chakraborty, and N. Ramakrishnan, "The deshredder: A visual analytic approach to reconstructing shredded documents," in *Proc. IEEE Int. Conf. Vis. Anal. Sci. Technol.*, Oct. 2012, pp. 113–122.
- [4] H. Zhang, J. K. Lai, and M. Bäcker, "Hallucination: A mixed-initiative approach for efficient document reconstruction," in *Proc. Workshops 26th AAAI Conf. Artif. Intell.*, 2012, pp. 54–60.
- [5] D. Pöhler *et al.*, "Content representation and pairwise feature matching method for virtual reconstruction of shredded documents," in *Proc. 9th IEEE Int. Symp. Image Signal Process. Anal.*, Sep. 2015, pp. 143–148.
- [6] Darpa. (2011). *Darpa Shredder Challenge*. Accessed: Aug. 31, 2018. [Online]. Available: <http://archive.darpa.mil/shredderchallenge>
- [7] A. Skeoch, "An investigation into automated shredded document reconstruction using heuristic search algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. Bath, Bath, U.K., 2006, p. 107.
- [8] M. A. O. Marques and C. O. A. Freitas, "Document decipherment-restoration: Strip-shredded document reconstruction based on color," *IEEE Latin Amer. Trans.*, vol. 11, no. 6, pp. 1359–1365, Dec. 2013.
- [9] A. Ukovich and G. Ramponi, "Feature extraction and clustering for the computer-aided reconstruction of strip-cut shredded documents," *J. Electron. Imag.*, vol. 17, no. 1, p. 013008, 2008.

- [10] R. Ranca, "Reconstructing shredded documents," Ph.D. dissertation, School Inform., Univ. Edinburgh, Edinburgh, U.K., 2013.
- [11] M. Prandstetter and G. R. Raidl, "Combining forces to reconstruct strip shredded text documents," in *Hybrid Metaheuristics*. Berlin, Germany: Springer, 2008, pp. 175–189.
- [12] M. Prandstetter and G. R. Raidl, "Meta-heuristics for reconstructing cross cut shredded text documents," in *Proc. 11th Annu. Conf. Genetic Evol. Comput.*, 2009, pp. 349–356.
- [13] A. Sleit, Y. Massad, and M. Musaddaq, "An alternative clustering approach for reconstructing cross cut shredded text documents," *Telecommun. Syst.*, vol. 52, no. 3, pp. 1491–1501, 2013.
- [14] B. Biesinger, C. Schauer, B. Hu, and G. R. Raidl, "Enhancing a genetic algorithm with a solution archive to reconstruct cross cut shredded text documents," in *Proc. EUROCAST*, 2013, pp. 380–387.
- [15] D. Applegate, R. Bixby, V. Chvatal, and W. Cook. (2003). *Concorde: A Code for Solving Traveling Salesman Problems*. Accessed: Aug. 29, 2018. [Online]. Available: <http://www.math.uwaterloo.ca/tsp/concorde>
- [16] P. De Smet, J. De Bock, and W. Philips, "Semiautomatic reconstruction of strip-shredded documents," *Proc. SPIE*, vol. 5685, pp. 239–249, Mar. 2005. [Online]. Available: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5685/0000/Semiautomatic-reconstruction-of-strip-shredded-documents/10.1117/12.586340.full>
- [17] Y. Wang and D.-C. Ji, "A two-stage approach for reconstruction of cross-cut shredded text documents," in *Proc. 20th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2014, pp. 12–16.
- [18] H. Xu, J. Zheng, Z. Zhuang, and S. Fan, "A solution to reconstruct cross-cut shredded text documents based on character recognition and genetic algorithm," *Abstract Appl. Anal.*, vol. 2014, Jun. 2014, Art. no. 829602. [Online]. Available: <https://www.hindawi.com/journals/aaa/2014/829602/abs/>
- [19] G. Chen, J. Wu, C. Jia, and Y. Zhang, "A pipeline for reconstructing cross-shredded English document," in *Proc. 2nd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2017, pp. 1034–1039.
- [20] F. A. Andaló, G. Taubin, and S. Goldenstein, "PSQP: Puzzle solving by quadratic programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 385–396, Feb. 2017.
- [21] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 9–16.
- [22] J. Balme, "Reconstruction of shredded documents in the absence of shape information," Dept. Comput. Sci., Yale Univ., New Haven, CT, USA, Tech. Rep., 2007.
- [23] M. Prandstetter, "Hybrid optimization methods for warehouse logistics and the reconstruction of destroyed paper documents," Ph.D. dissertation, Fac. Comput. Sci., Vienna Univ. Technol., Vienna, Austria, 2009.
- [24] M. Prandstetter, "Two approaches for computing lower bounds on the reconstruction of strip shredded text documents," Technische Universität Wien, Institut für Computergraphik und Algorithmen, Tech. Rep. TR1860901, 2009.
- [25] C. Schauer, M. Prandstetter, and G. R. Raidl, "A memetic algorithm for reconstructing cross-cut shredded text documents," in *Hybrid Metaheuristics*. Berlin, Germany: Springer, 2010, pp. 103–117.
- [26] C. Schauer, "Reconstructing cross-cut shredded documents by means of evolutionary algorithms," M.S. thesis, Inst. Comput. Graph. Algorithms, Vienna Univ. Technol., Vienna, Austria, 2010.
- [27] Y.-J. Gong, Y.-F. Ge, J.-J. Li, J. Zhang, and W. H. Ip, "A splicing-driven memetic algorithm for reconstructing cross-cut shredded text documents," *Appl. Soft Comput.*, vol. 45, pp. 163–172, Aug. 2016.
- [28] J. Chen, D. Ke, Z. Wang, and Y. Liu, "A high splicing accuracy solution to reconstruction of cross-cut shredded text document problem," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19281–19300, Aug. 2018.
- [29] W. Morandell, "Evaluation and reconstruction of strip-shredded text documents," M.S. thesis, Inst. Comput. Graph. Algorithms, Vienna Univ. Technol., Vienna, Austria, 2008.
- [30] H.-Y. Lin and W.-C. Fan-Chiang, "Reconstruction of shredded document based on image feature matching," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3324–3332, 2012.
- [31] J. Perl, M. Diem, F. Kleber, and R. Sablatnig, "Strip shredded document reconstruction using optical character recognition," in *Proc. 4th Int. Conf. Imag. Crime Detection Prevention (ICDP)*, 2011, pp. 1–6.
- [32] T. Phientrakul, T. Santitewagun, and N. Hnoohom, "A linear scoring algorithm for shredded paper reconstruction," in *Proc. 11th IEEE Int. Conf. Signal-Image Tech. Internet-Based Syst. (SITIS)*, 2015, pp. 623–627.
- [33] S. Guo, S. Lao, J. Guo, and H. Xiang, "A semi-automatic solution archive for cross-cut shredded text documents reconstruction," in *Proc. Int. Conf. Image Graph.* Cham, Switzerland: Springer, 2015, pp. 447–461.
- [34] N. Xing, S. Shi, and Y. Xing, "Shreds assembly based on character stroke feature," in *Proc. 2nd Int. Conf. Comput. Sci. Comp. Intell. (ICCSIL)*, vol. 116, 2017, pp. 151–157.
- [35] N. Xing and J. Zhang, "Graphical-character-based shredded Chinese document reconstruction," *Multimedia Tools Appl.*, vol. 76, no. 10, pp. 12871–12891, 2017.
- [36] Y. M. Y. Hasan and L. J. Karam, "Morphological text extraction from images," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1978–1983, Nov. 2000.
- [37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [38] D. Applegate, W. Cook, S. Dash, and D. Espinoza. (2007). *QSOPT Linear Programming Solver*. Accessed: Aug. 30, 2018. [Online]. Available: <http://www.math.uwaterloo.ca/~bico/qsopt>
- [39] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.
- [40] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. 12th IAPR Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Oct. 1994, pp. 566–568.
- [41] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.
- [42] R. Jonker and T. Volgenant, "Transforming asymmetric into symmetric traveling salesman problems," *Oper. Res. Lett.*, vol. 2, no. 4, pp. 161–163, 1983.
- [43] R. Smith, "An overview of the tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Sep. 2007, pp. 629–633.
- [44] P. Saboia and S. Goldenstein, "Assessing cross-cut shredded document assembly," in *Proc. Iberoamerican Congr. Pattern Recognit.* Cham, Switzerland: Springer, 2014, pp. 272–279.
- [45] T. A. Nartker, S. V. Rice, and S. E. Lumos, "Software tools and test data for research and testing of page-reading OCR systems," *Proc. SPIE*, vol. 5676, pp. 37–47, Jan. 2005. [Online]. Available: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5676/0000/Software-tools-and-test-data-for-research-and-testing-of/10.1117/12.587293.full>
- [46] C. Boettiger, "An introduction to docker for reproducible research," *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015.

Thiago M. Paixão received the B.S. degree in computer science from the Universidade Federal de Minas Gerais, Brazil, in 2007, and the M.Sc. degree in computer science from the Universidade de São Paulo, Brazil, in 2010. He is currently pursuing the Ph.D. degree with the Universidade Federal do Espírito Santo in Brazil. He is currently a Professor with the Instituto Federal do Espírito Santo.

Maria C. S. Boeres received the B.S. degree in mathematics from the Universidade Federal Fluminense, Brazil, in 1988, and the M.Sc. degree in computer and systems engineering and the Ph.D. degree in production engineering from the Universidade Federal do Rio de Janeiro, Brazil, in 2002 and 1992, respectively. She is currently an Associate Professor with the Universidade Federal do Espírito Santo, Brazil.

Cinthia O. A. Freitas received the B.S. degree in civil engineering from the Universidade Federal do Paraná, Brazil, the M.Sc. degree in electrical engineering and industrial informatics from the Centro Federal de Educação Tecnológica do Paraná, Brazil, in 1990, and the Ph.D. degree in applied computer science from Pontifícia Universidade Católica do Paraná (PUCPR), Brazil, in 2001. She is currently a Full Professor and a Researcher in the post-graduate degree in law at PUCPR. Her research interests are law and technology, law and societies, electronic contracts, document image analysis, and forensic science.

Thiago Oliveira-Santos received the B.S. degree in computer engineering and the M.Sc. degree in informatics from the Universidade Federal do Espírito Santo (UFES), Brazil, in 2004 and 2010, respectively, and the Ph.D. degree in biomedical engineering from the Universität Bern, Switzerland. He is currently Professor with the UFES.