

Writing Order Recovery in Complex and Long Static Handwriting

Moises Diaz¹, Gioele Crispo², Antonio Parziale², Angelo Marcelli², Miguel A. Ferrer³ *

¹ Universidad del Atlántico Medio, Las Palmas de Gran Canaria, Las Palmas 35017 (Spain)

² DIEM, University of Salerno, 84084 Fisciano (SA) (Italy)

³ Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones, Universidad de Las Palmas de Gran Canaria, Las Palmas 35017 (Spain)

Received 22 October 2020 | Accepted 4 March 2021 | Published 19 April 2021



ABSTRACT

The order in which the trajectory is executed is a powerful source of information for recognizers. However, there is still no general approach for recovering the trajectory of complex and long handwriting from static images. Complex specimens can result in multiple pen-downs and in a high number of trajectory crossings yielding agglomerations of pixels (also known as clusters). While the scientific literature describes a wide range of approaches for recovering the writing order in handwriting, these approaches nevertheless lack a common evaluation metric. In this paper, we introduce a new system to estimate the order recovery of thinned static trajectories, which allows to effectively resolve the clusters and select the order of the executed pen-downs. We evaluate how knowing the starting points of the pen-downs affects the quality of the recovered writing. Once the stability and sensitivity of the system is analyzed, we describe a series of experiments with three publicly available databases, showing competitive results in all cases. We expect the proposed system, whose code is made publicly available to the research community, to reduce potential confusion when the order of complex trajectories are recovered, and this will in turn make the trajectories recovered to be viable for further applications, such as velocity estimation.

KEYWORDS

Cluster Resolution, Complex and Long Handwriting, Good Continuity Criteria, Writing Order Recovery.

DOI: 10.9781/ijimai.2021.04.003

I. INTRODUCTION

OVER the last 40 years, handwriting analysis and recognition have been widely studied, and many theoretical and experimental results have been obtained both on handwriting acquired with tablets (on-line samples) and on that obtained with scanners (off-line samples) [1], [2]. These studies have contributed to many useful applications, such as mail sorting, form processing and handwriting recognition. More recently, the widespread use of devices such as smartphones, tablets, and electronic pen pads has given rise to a personal digital bodyguard concept [3]. This feature can supplement data protection, which enhances human-machine interactions through handwriting recognition.

In addition to such technological advances, the automatic processing of off-line handwriting is still of great interest in many fields of application, such as enterprise management [4],[5], education [6]–[8], and healthcare [9], [10]. Both public offices and private companies need to archive and retrieve digital versions of documents that contain handwritten samples. Recent years have witnessed the rise of many

digital libraries that require systems that allow automatic searches in transcripts and ancient manuscripts [11].

It has long been understood that on-line handwriting systems perform much better than off-line ones because the former have access to dynamic information. This information consists mainly of the order in which the trajectory was written and its velocity profile. This limitation has motivated the development of systems for writing order recovery in static images [12]–[14]. These systems include two steps: 1) thinning of the handwriting, and 2) writing order recovering of the trajectory. Computer-based systems for recovering static trajectories have been proposed for several handwritten applications, including the reading of cursive handwriting [15], Latin and Arabic handwriting word recognition [16]–[18], Indian and Chinese character recognition [19], [20], digit recognition [21], historical document transcription [22], mathematical symbol recognition [23], signature verification [14], [24], [25], writer verification and identification [26], [27], handwriting style modeling [28], [29] and handwriting analysis and synthesis [30].

Recovering the trajectory in static handwriting may lead to multiple possible solutions, with only one representing the real trajectory sought. Solving this inverse problem becomes even more complicated when the handwritten pattern contained in the image comprises multiple components;¹ in this case, recovering a new component may

* Corresponding author.

E-mail addresses: moises.diaz@atlanticomedio.es (M. Diaz), crispiogioele@gmail.com (G. Crispo), anparziale@unisa.it (A. Parziale), amarcelli@unisa.it (A. Marcelli), miguelangel.ferrer@ulpgc.es (M.A. Ferrer).

¹ By “component”, we refer to a piece of writing with two end-points, meaning that it is performed between two pen-ups. It can be denoted as a pen-down as well.

Please cite this article in press as:

M. Diaz, G. Crispo, A. Parziale, A. Marcelli, M. A. Ferrer. Writing Order Recovery in Complex and Long Static Handwriting, International Journal of Interactive Multimedia and Artificial Intelligence, (2021), <http://dx.doi.org/10.9781/ijimai.2021.04.003>

depend on the order of the previous one and the in-air trajectory between two consecutive components.

A. Our Contribution

This paper proposes a novel system for estimating the writing order recovering in complex and long static handwriting consisting of many components, separated by pen-ups, also named in-air trajectories. Inspired by human movements during the production of handwriting and by motor control perspectives [31], our system chooses the smoothest ballistic trajectories based on good continuity criteria when writing or drawing. To this end, we apply some multiscale strategies to recover the trajectory order in the agglomeration of pixels, which we call clusters. Additionally, we study the proposed system's performance, both when the end-points of the components are known and when they are unknown. We provide a quantitative measure of how heuristics on the choice of starting and ending points affect the writing order recovery. The effectiveness of the system is demonstrated in three databases, achieving competitive performance.

We focus our work on static specimens for two reasons: the first one is that developing a system that recovers complex, long, and discontinuous trajectories represent the most challenging case, and the second is that estimating the writing order from static images is of great interest for the early prescreening of neurodegenerative disorders [10], [32].

The code is developed in Matlab, and it is freely provided for research².

The outline of the paper is as follows: Section II presents a brief overview of related works on writing order recovery of trajectories in static handwritten specimens. Section III describes the proposed solution for estimating handwriting recovering. A sensitivity and stability study of parameters is given in Section IV, whereas experimental results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. LITERATURE REVIEW

Over the last thirty years, many systems have been proposed for recovering static trajectories, with two surveys assessing the related state of the art [12], [13]. Moreover, a competition has been considered in order to establish a common benchmark for writing order recovery of Arabic signatures [14], [33].

Systems used to recover trajectories from static handwriting employ three main approaches: contour-based, skeleton-based, and learning-based. The first two differ in terms of the "object" used to represent the handwriting, respectively its contour and its skeleton. The skeleton is the result of a thinning process that produces a 1-pixel-wide line, which follows the centerline of the original image, and that ideally corresponds to the original pen-tip trajectory.

Regarding contour-based approaches, [34] describes a method that segments cursive handwriting by detecting the points where the trace contour has the maximum curvature. The segments are ordered based on the contour curvature smoothness. Another handwriting segmentation is given in [16], and proceeds through an analysis of the handwritten contour. Here, the writing order trajectory is estimated by adopting graph-based representations at both the segment and stroke levels. The list of candidate paths is obtained by choosing curvature and width stroke preservation, which are local continuity criteria, as cost functions. Following up on [16], the system was employed in [17] to develop an off-line recognition system. In that work, a Hidden Markov Model (HMM) was trained for selecting the most likely written trajectory from the list of candidate paths. To avoid

introducing artifacts in the handwriting, preprocessing steps, such as binarization, were omitted in [35]. Instead, the authors extracted control points from the grayscale images, which were used to recover the trajectory according to certain heuristics rules.

The skeleton-based methods can be categorized as local line order recovering and global graph searching methods. The former reconstructs the trajectory by choosing the most plausible direction at each branch point of the skeleton. These methods are simple and have a low computational cost, but their performances are typically limited by the difficulty of designing heuristic rules for different handwriting styles. Instead, graph searching methods recover the trajectory by representing the topological structure of the skeleton with a graph and traversing it. They have a greater computational cost and their performances depend on the definition of criteria for selecting the best trajectory among many alternatives.

An example of local line recovering is given in [36], where the trajectories are recovered according to good continuity criteria, which take into account the direction, length, and width of the strokes making up characters. Off-line signatures are recovered in [37] by following heuristic rules inspired by the way human beings write a given shape. The rules are applied to deal with low-level pixel processing and high-level stroke processing. A similar approach is applied to handwritten digits in [38]. In [39], a taxonomy of local, regional, and global features that can be used for recovering temporal properties from the image is proposed. On the other hand, a likelihood measure is developed in [40]. It selects the most likely writing order recovery from the analysis of skeletons. In an on-line automatic signature verification system, the method proposed in [40] shows a greater false acceptance rate in skilled forgeries than in random forgeries. A genetic algorithm is also used for recovering segments extracted from the skeleton [41]. The fitness function adopted by the genetic algorithm for selecting the best individual of the population takes the writing direction, the repetition of segments, and the angular deviation on the crossing of the occlusion stroke into account. Additionally, a complete framework to recover the dynamic properties (i.e., velocity and pressure) from an image-based signature is presented in [24] by using classical approaches to recover the static trajectories in the signatures.

As for global graph methods, they represent the topological structure of the skeleton with a graph whose vertices represent the end-points, the junctions and the contact points of the skeleton, and whose arcs represent lines and curves. These methods determine the writing order recovery by finding the most appropriate path along the graph. In [42], one of these methods is proposed for recovering the trajectory of words through the traveling salesman problem. Based on the skeleton of words, the authors search for trajectories with minimum curvatures. Also, in [43], each component of a word is represented by a graph. The trajectory of the whole word is obtained by concatenating, from left to right, the most likely trajectory of each component. The authors define both global and local criteria that allow to select the best trajectory. The path search was based on the best-first search algorithm. The authors in [44] propose to construct a graph to use to determine the types of each edge from the skeletons. Then, they develop a writing order recovery algorithm to traverse such a graph without applying any graph search algorithm. The goal of the method proposed in [45] is to use as little heuristic knowledge as possible. To this end, the proposal applies the maximum weighted matching of general graphs to find double-written lines. It exploits the minimum energy cost criterion as a guiding principle for recovering the trajectories. Successful results in single-stroke images are obtained.

A graph transformation is proposed in [46] to ensure that all graphs' nodes had an even number of incident arcs. This property allows to traverse the graph by using Fleury's algorithm, combined with handwriting generation models. The approach exhibited a

² github.com/gioelecrispo/wor

reduced computational cost because it divided the whole graph into sub-graphs by detecting the graphs' bridges. This method is improved in [47] by introducing a feedback connection between the unfolding module and a module that extracts elementary movements from the recovered trajectory. The method exploits the analogies between unfolding and segmentation processes and those occurring in the brain when a trajectory plan is learned and executed.

Furthermore, the search for the optimal trajectory along the graph representing the skeleton of a word is executed in [48]. The authors use both the Greedy and the Dijkstra algorithms with a well-defined smoothness function. Further, they focus on reconstructing the optimal trajectory of words by splitting them into many strokes according to their respective curvature values and a set of rules of handwriting. Good continuity criteria derived from both visual perception and movement execution are applied to signatures in [50]. In particular, the implementation focuses on a multiscale analysis of the thinned trajectories and the Dijkstra algorithm.

For learning-based approaches, they require some exemplars of static images with the corresponding drawing orders. Thus, models are trained to recover the trajectories of new static images. In [51], an HMM is adopted for recovering the trajectory of single-stroke handwritten signatures. Each state of the HMM has a probability density function that embeds geometric shape information of the static image, while transition probabilities define the possible pen movements between static image coordinates. A training phase is proposed in [52], in which the original trajectory order and other attributes, such as the length and direction, are extracted from a set of on-line scripts to build a universal writing model. Then, it is used to reconstruct the drawing order during the test phase. The skeleton of the static image is matched to the model by using a dynamic programming algorithm, and the trajectory with maximum likelihood is selected.

Since 2018, deep learning techniques have been exploited in recovering trajectories in Chinese, Japanese, Indic, Arabic, and Latin characters (e.g. [41], [58]). While these techniques may be promising, they have the disadvantage of requiring a huge amount of data for training. Presently, deep learning methods are used to estimate the trajectory of characters and numbers, which are less complex than words and signatures. In [53], an algorithm is proposed based on a regression Convolutional Neural Network (CNN) model to predict the probability of the next stroke point position. The same authors present improvements of such an architecture in [54], based on two CNNs. Another model based on an encoder-decoder LSTM module was introduced in [20]. Here, the encoder module consists of a Convolutional LSTM network, which takes an off-line character image as input and encodes the feature sequence to a hidden representation. The output of the encoder is fed to a decoder LSTM that sequentially predicts the coordinate points. The architecture is tested on characters from three Indic scripts. Experimentation shows that the main limitation of the approach is the need to train a separate model for each individual script.

A handwriting recognition system based on a writing order recovery algorithm is proposed in [55]. The order recovery algorithm exploits an end-to-end system based on a VGG-LSTM, which extracts

and encodes features, followed by a BLSTM used as a decoder to generate temporal coordinates. The method could eventually produce human-like velocities [18]. Moreover, a network of two variational auto-encoders is proposed in [56] to convert on-line and off-line handwritten Latin characters to each other. An improved VGG-16 CNN model is proposed in [57] to recover the handwriting stroke order. The CNN model recovers the writing order effectively, even if the accuracy of the network decreases as the number of strokes increases.

As summarized in Table I, an analysis of the state of the art shows that the static handwriting trajectory recovering problem is far from solved despite the very high number of papers that have been published on the subject. Many methods have been designed, and they mostly start from the assumption that the static image consists of a single component, i.e., a pen-down. This assumption does not hold in real applications, where handwriting patterns consist of many pen-downs and pen-ups, such as in signatures. Furthermore, many systems on the subject have not been validated on public datasets, but rather, have only been tested on a few samples. Even more, performance of some systems are only qualitatively measured. Eventually, we notice that studies reported in Table I adopt different metrics and strategies for measuring the goodness of the writing order recovery systems: some authors use the accuracy obtained by recognition systems on the recovered trajectory as an indirect metric, whereas others compare the recovered trajectory with the data acquired by a tablet.

III. ESTIMATING THE WRITING ORDER IN HANDWRITING

The objective of the proposed system is to estimate the writing order of the 8-connected thin line representation of the handwriting image. The analysis of two elements (components and clusters) plays a vital role in the trajectory recovering process. A cluster is formed following the intertwining of different strokes, resulting in an agglomeration of pixels. Each pixel in the cluster therefore has more than two pixels connected in its 8-neighborhood, and consequently, the clusters complicate the accurate component drawing process.

A flowchart of the proposed system is depicted in Fig. 1. It is composed of three stages. 1) *Point classification*, where the clusters of a thinned specimen are identified, 2) *Local examination*, where individual clusters are resolved by joining their output branches, and 3) *Global reconstruction*, where we estimate the handwriting order with multiple pen-downs. The mathematical notation used in this work is provided in Table II.

A. Point Classification

We classify each point (or pixel) of the thinned trajectory into one of three categories: (1) end-points, which are pixels with only one 8-neighbor, (2) trace points which have two 8-neighbors, and (3) branch points which have three or more 8-neighbors. Fig. 2 gives a visual example of this categorization. To this end, we identify the clusters as the sets of adjacent pixels labeled as branch points.

To classify the black pixels in the image, we search for their connectivity in their 8-neighbors. The complexity order of this procedure is $\mathcal{O}(8 \cdot h \cdot w)$, where (h, w) are the height and width of

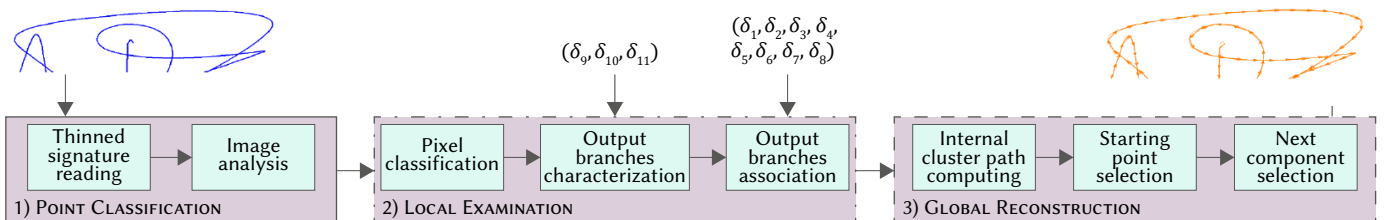


Fig. 1. Flowchart of the proposed system for writing order recovering.

TABLE I. LITERATURE REVIEW ON SYSTEMS FOR RECOVERING STATIC TRAJECTORIES

Ref.	Approach	Pattern	Dataset	Evaluation
[34]	Contour-based	Words	Private. 200 images of words written by six different writers.	Quality of ordering ranked by human subjects.
[16], [17]	Contour-based	Words	Public. 10,448 words taken from the IRONOFF dataset. For each word, the on-line and off-line versions are available.	Accuracy of HRS.
[35]	Contour-based	Signatures	Public. SVC 2004 dataset. Images are synthesized from on-line samples.	Preliminary study. The number of detected stroke points.
[36]	Skeleton - Local line	Characters	Private. 10,000 characters written 20 subjects. 2/3 of samples used as a test set.	Quality of ordering ranked by human subjects.
[37], [38]	Skeleton - Local line	Signatures and Numerals	Private. 20 signatures [37] and 150 numerals [38].	Visual inspection for signatures and HRS for numerals.
[39]	Skeleton - Local line	Words and Numerals	Private. 1000 images from US mail service.	Visual inspection.
[40]	Skeleton - Local line	Signatures	Public. The last fifty users of the MCYT-100 dataset wrote the signatures. Images are synthesized from on-line samples.	FAR and FRR of ASV.
[41]	Skeleton - Local line	Words	Public. Hundreds of samples from LMCA, IRONOFF, and IFN/ENIT datasets.	Visual inspection.
[24]	Skeleton - Local line	Signatures	Public. Fifty users of the BiosecurID dataset. Images are synthesized from on-line samples.	EER of ASV.
[42]	Skeleton - Graph-based	Characters	Private.	Visual inspection.
[43]	Skeleton - Graph-based	Words	Private. 150 words written by five subjects.	Accuracy of HRS.
[44]	Skeleton - Graph-based	Words	Private. 100 images.	Visual inspection.
[45]	Skeleton - Graph-based	Characters	Public. 708, 811 images obtained by converting on-line data of the Unipen dataset.	Rate of correct complete trajectory recovery, Accuracy of HRS.
[46],[47]	Skeleton - Graph-based	Words	Private. 6500 images containing cursive handwriting.	Similarity measure between automatic and manual drawing order.
[48]	Skeleton - Graph-based	Words and Characters	Private. 6868 images taken from 3 different datasets. Public. Images were taken from IRONOFF [49].	RMSE, DTW, Accuracy of HRS.
[50]	Skeleton - Graph-based	Signatures	Public. 1953 and 2820 on-line signatures from the SigComp2009 and SUSIG - Visual datasets, respectively. The thinned version of the on-line signatures is used.	RMSE, DTW, Number of clusters correctly solved.
[51]	Learning-based	Signatures	Public. 710 single-stroke on-line signatures from fifty users taken from US_SIGBASE and Dolfig datasets. Images are synthesized from on-line samples.	Accuracy score that measures the alignment of the recovered order and the ground truth.
[52]	Learning-based	Signatures	Private. 300 images of signatures.	The rank of the proposed recovering trajectories.
[53], [54]	Learning-based	Characters and digits	Public. OLHWDB 1.1 dataset containing 3755 Chinese characters; 2000 English letters and Arabic digit symbols in UNIPEN.	Rate of correct: end point selection [54], branch points resolution [54], complete trajectory recovery [53], [54].
[20]	Learning-based	Characters	Public. LIPI Toolkit dataset (Tamil, Telugu and De-vanagari characters). Around 21, 000 characters per script.	Rate of correct: starting point selection, junction points resolution, and complete trajectory recovery.
[55]	Learning-based	Characters and digits	Public. LMCA and IRONOFF datasets.	Accuracy of HRS.
[56]	Learning-based	Characters	Public. Unipen.	DTW.
[57]	Learning-based	Characters	Public. OLHWDB 1.1 dataset.	Rate of correct complete trajectory recovery.

HRS: Handwriting Recognition System, ASV: Automatic Signature Verifier.

TABLE II. NOTATION USED

Notation	Description
r	Rank of the cluster
c_{ij}	Curvature between output branches i and j
α_i, β_i	External and internal angle for branch i
π_{ij}	Weighted angle direction between branches i and j
$\omega_{ext}, \omega_{int}, \omega_{cur}$	External, internal and curvature weights for computing the anchor point direction
$\delta_1, \delta_2, \delta_3$	Parameters for identifying retracing
$\delta_4, \delta_5, \delta_6$	Parameters for identifying T-patterns
δ_7, δ_8	Parameters for identifying coupled clusters
δ_9	Number of trace points
δ_{10}	Parameter for the brotherhood
δ_{11}	Number of points for computing the curvature

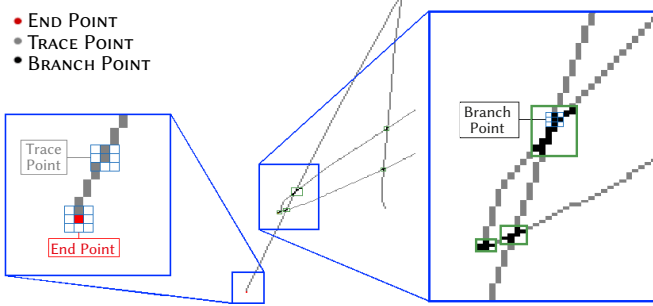


Fig. 2. Point classification in a thinned trajectory (center of the image): details inside the blue rectangles with clusters inside the green ones.

the image, respectively. Next, the clusters are defined as the sets of adjacent pixels previously labeled as branch points. The clusters are thus identified by performing the connected component labeling algorithm over the branch points. As this algorithm requires at most two scans of the image, its computational complexity is $\mathcal{O}(2 \cdot h \cdot w)$.

B. Local Examination on the Clusters

Recovering a skeleton from an end-point to another through the trace points is an effortless operation. However, the task becomes more demanding when clusters are encountered, since we must then decide on the adequate output branch to recover. The output branches can be also defined as all the trace points that converge in a single cluster.

For this writing order recovery, given a cluster, firstly, its pixels are classified. Secondly, the output branches are characterized, and, finally, they are paired off in input-to-output paths.

C. Pixel Classification in a Cluster

The pixels inside a cluster are classified according to their connectivity as follows:

- **Cluster points.** These are branch points.
- **Anchor points.** These are branch points of the clusters having at least one trace point (i.e., a point outside the cluster) as a neighbor. Since the cluster output branches are anchored on them, they are denoted as *anchor points*. Moreover, the number of anchor points establishes the rank of the cluster, denoted by r .
- **False trace points.** These are labeled as cluster points whether at least one of the following conditions hold: (1) they are connected

to two different cluster points of the same cluster; (2) they are connected to a cluster point and to a false trace point of the same cluster; (3) they are connected to two false trace points of the same cluster.

Such a classification in a single cluster requires a computational complexity of $\mathcal{O}(p^2)$, with p being the total number of cluster points. Fig. 3 shows an example of a cluster and its classified pixels.

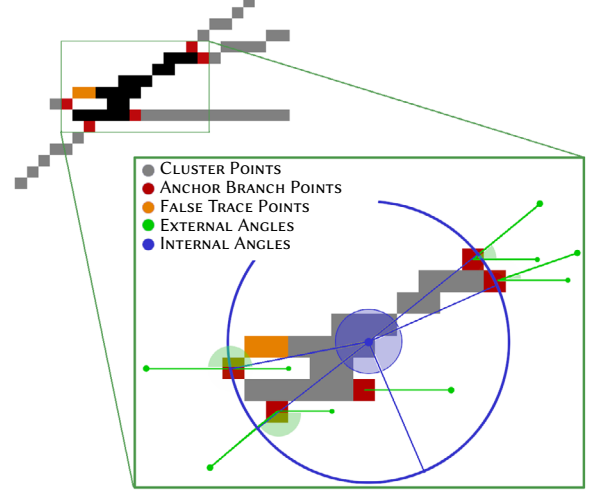


Fig. 3. Example of a cluster: cluster points (in gray), anchor points (in red), false trace points (in orange), external angle directions (in green), and internal angle directions (in blue).

D. Output Branches Characterization

Cluster points and false trace points are identified through their horizontal and vertical coordinates in the images. Beyond the Cartesian coordinates, the anchor points are characterized through the angle direction of the output branches. We developed three strategies for computing the direction of the output branches.

- **External angle.** A multiscale approach [59] is devised to compute this angle by using only information external to the cluster. This algorithm considers the $\{x_i, y_i\}_{i=1}^{\delta_9}$ coordinates of an anchor point (AP_x, AP_y) and δ_9 numbers of trace points of the output branch (see Table II). The output of this algorithm is the external angle $\alpha_i, \forall i \in 1, \dots, r$. The algorithm is formalized in Algorithm 1, which runs in time $\mathcal{O}(\delta_9^2)$. Additionally, Fig. 3 illustrates the external angles on the anchor points.
- **Internal angle.** The internal angles can compensate for some shortcomings of the external angles when the output branches contain a few trace points. For this compensation, the cluster center of gravity of the anchor points is calculated. The (x, y) coordinates of the first δ_9 trace points from an output branch are considered and the internal angles $\beta_i, \forall i \in 1, \dots, r$ are computed by the procedure reported in Algorithm 2, its complexity being $\mathcal{O}(\delta_9^2)$.
- **Curvature.** We combine all pairs of output branches, $C_2^r = \binom{r}{2}$, chosen among the r available anchor points in a cluster. The continuity connection of two output branches (i, j) and its shortest path is obtained through Dijkstra's algorithm [60]. Each of the 8-connected traces thus created is divided into n equidistant points and a curvature representative value is calculated ($\rho = [\rho_1, \rho_2, \dots, \rho_n]$). Finally, the maximum of the difference between adjacent elements of ρ is used to quantify the curvature of the trace generated with the output branches (i, j) . The curvature c_{ij} goes from 0, if the curve is perfectly straight, to 180 degrees, if the curve is bent over itself. The procedure is detailed in Algorithm 3, which has a complexity of $\mathcal{O}(2\delta_9^2)$.

Algorithm 1. Compute External Angle

```

1: procedure COMPUTEEXTANG( $AP_x, AP_y, x, y$ )
    $\triangleright \delta_9$  being the scale of the multiscale approach and the length of  $(x, y)$ 
2: for  $s = 1$  to  $\delta_9$  do
3:    $k \leftarrow 1$ 
4:    $nextPixel \leftarrow (AP_x, AP_y)$ 
5:   for  $i = 1$  step  $s$  to  $\delta_9$  do
6:      $distX \leftarrow nextPixel_1 - x_i$ 
7:      $distY \leftarrow nextPixel_2 - y_i$ 
8:      $angInt_k \leftarrow atan2(distY, distX)$ 
9:      $k \leftarrow k + 1$ 
10:     $nextPixel \leftarrow (x_i, y_i)$ 
11:   end for
12:    $minAngle \leftarrow \min$  of  $angInt$ 
13:    $maxAngle \leftarrow \max$  of  $angInt$ 
14:    $angMS_{(s)} \leftarrow \text{linear interpolation}$ 
15: end for
16: for  $s = 1$  to  $\delta_9$  do
17:    $resMultiscale \leftarrow \text{circular mean of } angMS_{(s)}$ 
18: end for
19:  $\alpha \leftarrow \text{circular mean of } resMultiscale$ 
20: return  $\alpha$ 
    
```

E. Handling Special Cases: the Brotherhood

The brotherhood refers to a set of joined clusters. Two clusters can be merged when there are less than δ_{10} trace points from each other, and are connected by at least two branches. The brotherhood is a full-fledged cluster, and we apply the same rules of a single cluster to it. Fig. 4 shows two examples of brotherhood process. It is worth pointing out that cluster C3 (Fig. 4a) and cluster C1 (Fig. 4b) are not included in the brotherhood because they are connected with the other clusters only through one branch.

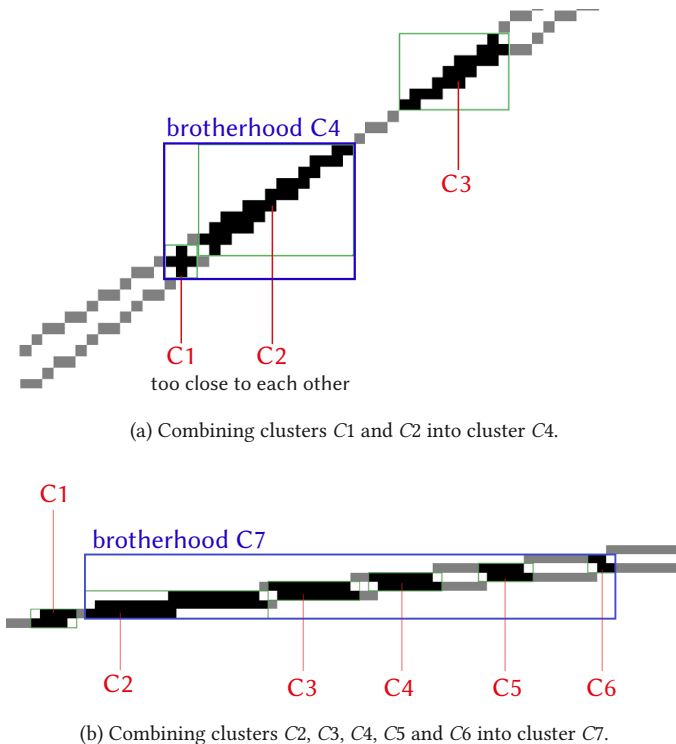


Fig. 4. Examples of two brotherhood application.

Algorithm 2. Compute Internal Angle

```

1: procedure COMPUTEINTANG( $ccg_x, ccg_y, x, y$ )
    $\triangleright \delta_9$  being the length of  $(x, y)$ 
2: for  $i = 1$  to  $\delta_9$  do
3:    $vx_i \leftarrow ccg_x - x_i$ 
4:    $vy_i \leftarrow ccg_y - y_i$ 
5:    $ang_i \leftarrow atan2d(vx_i, vy_i)$ 
6: end for
7:  $\beta \leftarrow \text{circular mean of } ang$ 
8: return  $\beta$ 
    
```

Algorithm 3. Compute Curvature

```

1: procedure COMPUTECURV( $x, y, \delta_{11}$ )
2:    $m \leftarrow \text{length of } (x, y)$ 
3:    $n \leftarrow \min(\delta_{11}, m)$ 
4:   if  $m \leq 2$  then
5:     return 0  $\triangleright$  The curve is straight
6:   else
7:      $s \leftarrow \text{floor}(\text{linspace}(1, m, n))$ 
    $\triangleright$  Indexes of  $(x, y)$  to be considered
8:   for  $l = 1$  to  $n$  do
9:     for  $a = 1$  to  $\delta_{11}$  do
    $\triangleright$  Forward points
10:    if  $l \geq 1$  and  $l \leq n - 1$  then
11:       $k \leftarrow \min(n - l, a)$ 
12:      for  $f = 1$  to  $k$  do
13:         $v_f \leftarrow \text{atan2d}(y_{s_l} - y_{s_{l+f}}, x_{s_l} - x_{s_{l+f}})$ 
14:      end for
    $\triangleright$  Backward points
15:    if  $l \geq 2$  and  $l \leq n$  then
16:       $k \leftarrow \min(l - 1, a)$ 
17:      for  $b = 1$  to  $k$  do
18:         $v_b \leftarrow \text{atan2d}(y_{s_l} - y_{s_{l-b}}, x_{s_l} - x_{s_{l-b}})$ 
19:      end for
20:       $v \leftarrow \text{concat}(v_b, v_f)$ 
21:       $ang_a \leftarrow \text{circular mean of } v$ 
22:    end for
23:     $\rho_l \leftarrow \text{circular mean of } v$ 
24:  end for
25:   $c = \max(\Delta\rho)$ 
26:  end c
    
```

The complexity of the brotherhood is $\mathcal{O}(nC \cdot r \cdot \delta_{10}^2)$, where nC is the number of clusters in the brotherhood, r the rank of the cluster and δ_{10} the number of trace points.

F. Output Branches Association

The Gestalt theory [61] states that all elements of sensory input are perceived as belonging to a coherent and continuous whole. Moreover, such criteria are supported by the studies of motor control theories [62], in particular those related to the execution of rapid and smooth movements that involve the principle of energy minimization. Under these perspectives, good continuity criteria are taken into account.

As such, we pair the two exit directions whose external difference is closer to 180 degrees than are the others, remove them from the cluster, and repeat the pairing until the rank of the cluster is either zero (meaning there are no more exit directions to pair) or three. Let

i and j be two branches of a cluster. To implement this criterion, the following weighted angle direction, $\pi_{i,j} \forall (i,j) \in r$, is calculated:

$$\pi_{i,j} = \omega_{ext} \cdot |\alpha_i - \alpha_j| + \omega_{int} \cdot |\beta_i - \beta_j| + \omega_{cur} \cdot c_{i,j} \quad (1)$$

where (α_i, α_j) refer to the external angles, (β_i, β_j) to the internal angles and $c_{i,j}$ denotes the curvature between the considered branches. The weights in the angles and curvature, $(\omega_{ext}, \omega_{int}, \omega_{cur})$ (see Table II), have to satisfy that the sum of their values be equal to one, with each weight ranging in the $(0, 1)$ interval. According to eq. (1), the smaller $\pi_{i,j}$ the smoother the line connecting the i -th and j -th output branches. Once all the $\pi_{i,j}$ are calculated, we process the clusters depending on their ranks:

- for even-rank clusters, we select the pair corresponding to the smallest values of $\pi_{i,j}$ and removing the paired branches. These steps are repeated $\frac{r}{2}$ times, until all the branches are paired;
- for odd-rank clusters, the same procedure as above is applied $\frac{r-1}{2}$ times. The remaining branches constitute a 3-rank 2 cluster.

The 3-rank clusters are by far the toughest to manage [45],[63]. They can be classified according to their geometrical and morphological properties (see Fig. 5) as follows:

- **T-pattern clusters:** These are clusters whose shape is similar to a ‘‘T’’. With these, one out of the three angles $\pi_{i,j}$ had to satisfy $180(1 - \delta_4) \leq \pi_{i,j} \leq 180(1 + \delta_4)$, and the other two, the following condition: $\pi_{i,j} \frac{100}{360} \leq \delta_5$. To avoid 360 misclassification, any branch of a T-pattern cluster should not be too close to an end-point and to another 3-rank cluster. Let dep be the distance in pixels between an anchor branch point and the nearest end-point and let d^{3rc} be the distance between an anchor branch point and the nearest 3-rank cluster. Then, the last condition to satisfy to consider a T-pattern would imply: $\min(d^{ep}, d^{3rc}) \geq \delta_6$.
- **Retraced clusters:** These appear in closed handwritten loops, and consequently, at least one of the branches has to end in an end-point. Let d_k^{ep} be the minimum distance in pixels from the branch k to its end-point. Then a retracing should satisfy $d_k^{ep} \leq \delta_2$. Also, as the candidate retraced branch is expected to be straight, the value of its curvature c_k should be less than or equal to δ_3 . Finally, the opposite angle $\pi_{i,j}$ should also satisfy: $\frac{100\pi_{i,j}}{360} \leq \delta_1$.
- **Coupled clusters:** These are two-neighbor 3-rank clusters, none of which is a T-pattern or a retraced cluster, which share one output branch whose length $d_{ab} \leq \delta_7$. As such, the other branches of both clusters compose a 4-rank cluster, which has to respect the following relation:

$$\max[\text{avg}(\pi_{1,3}, \pi_{2,4}), \text{avg}(\pi_{2,3}, \pi_{4,1}), \text{avg}(\pi_{2,1}, \pi_{3,4})] \leq \delta_8 \quad (2)$$

This condition means that the good continuity criterion must be strongly observed for these clusters to be classified as coupled.

- **Normal cluster:** This is a 3-rank cluster that does not belong to the previous classes. In this case, the branches corresponding to the smallest values of $\pi_{i,j}$ are associated, and the remaining one is disjointed from the cluster.

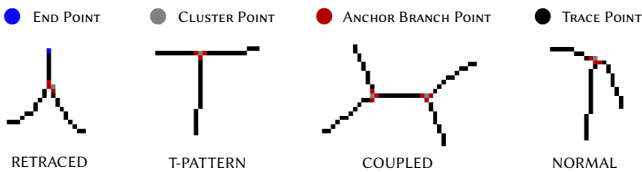


Fig. 5. Classification of the 3-rank clusters.

In our implementation, firstly, we pair the branches in even rank clusters, because these clusters are the less ambiguous ones. Secondly,

we work out the odd cluster whose rank is higher than three. Finally, we deal with the three rank clusters since they are the most confusing type.

G. Global Reconstruction

The goal of this section is to recover the writing order of the pen-downs. To this end, we need to define a path that traverses the cluster to link two previously paired branches. Moreover, the starting points of each component and their order need to be estimated as well.

H. Computing Internal Cluster Paths

Once the anchor branch points are paired, we define the path that connects them. Let a cluster be composed of p pixels. The adjacency matrix A is then a $p \times p$ matrix defined as follows: Given a pair of neighbor pixels (p_i, p_j) within the cluster, $A(i, j)$ would have a value of two or three. If (p_i, p_j) are not neighbors or $p_i = p_j$, the value in the adjacency matrix is zero. Based on observations in trajectory generation [31], the weights assigned operate in compliance with the principle of minimizing energy. Accordingly, to connect two neighboring pixels, we exercise a preference in choosing a straight connection rather than an oblique one. At the same time, an oblique connection is preferred against two straight connections.

The adjacency matrix A is therefore processed by the Dijkstra algorithm [60], which approaches the Gestalt theory perspective of good continuity [61] and rapid movement trajectories.

I. Starting Point Selection

To choose the starting point, we model the spatial distribution of the starting points with a two-dimensional Gaussian function.

To this end, we store the starting end-point of a random number of handwriting. The median of this Gaussian function is experimentally located at $0.15 \cdot h$ and $0.35 \cdot w$, with h and w being the height and the width of the writing area on the top-left part of the images. When there are no end-points within the ellipse defined by the mean and the two standard deviations of the Gaussian, the leftmost end-point is selected as the starting point, as usual in Western handwriting.

J. Next Component Selection

Once the first component is recovered, the next component is chosen according to a proximity criterion; this is the component with the nearest end-point, which has not yet been recovered. Formally, this criterion can be described as:

$$i^* = \arg \min_{i,j} \left(\sqrt{(x_{ep_i} - x_{ep_j})^2 + (y_{ep_i} - y_{ep_j})^2} \right) \quad (3)$$

where (x_{ep_i}, y_{ep_i}) are the coordinates of the last recovered end-point and (x_{ep_j}, y_{ep_j}) the coordinates of the end-points of components that have not yet been recovered.

Finally, Fig. 6 illustrates three examples obtained with the proposed method.

IV. EXPERIMENTAL SENSITIVITY AND STABILITY ANALYSIS

In this section, we analyze the sensitivity and the stability of our system in terms of the values of the parameters listed in Table II, δ_1 to δ_{11} , ω_{ext} , ω_{int} , and ω_{cur} , which have been determined heuristically. The performance of the system depends on the parameter values of the system. We define the accuracy rate θ as a performance measure: $\theta = \frac{\#clusters \text{ correctly solved}}{\#total \text{ clusters}}$.

A cluster is counted as *correctly solved* if all its branches are paired similarly to the real trajectories, which are used as ground truth data.

The parameters and weights were heuristically optimized in a trial and error procedure in three steps. Initially, all the weights were set to 0.33 and a coarse tuning of the parameters δ_k was conducted.

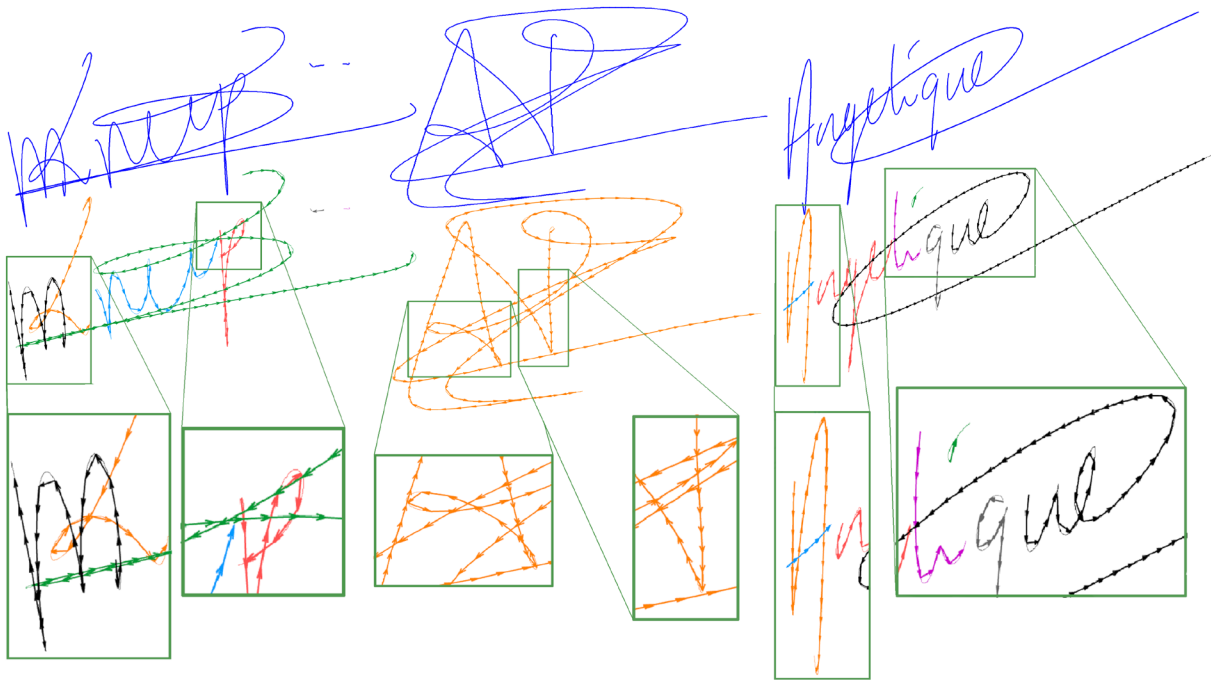


Fig. 6. Examples of writing order recovering in signatures with the proposed method. The recovered writing order of each component is represented by directional arrows, whereas different colors in the estimated trajectories represent different detected components.

Then, the performance of θ was analyzed independently for each decision criterion, cluster rank and type to fine-tune the parameters. Eventually, the weights were fine-tuned for a final maximization of θ .

To investigate whether the heuristic optimization of parameters was biased, we carried out both a sensitivity and a stability study. In both cases, our optimization procedure was performed with 30 % of the specimens randomly selected from the SigComp2009 database [64]. Each experiment was repeated ten times with different values of each parameter under investigation, and the corresponding average accuracy rates were reported.

A. Sensitivity Study: Results

In the study, the *heuristic values* of the parameters δ_k were individually varied, and the results given in terms of the *sensitivity grade*, defined as $\Delta\theta/\Delta\delta$.

TABLE III. VALUES OF δ_k PARAMETERS AND THEIR VARIABILITY RANGE FOR THE SENSITIVITY STUDY

δ_k	Value	Type	Range
δ_1	28	ND*	(22, 34)
δ_2	20	ND	(10, 30)
δ_3	20	degrees	(10, 30)
δ_4	3	ND	(1, 5)
δ_5	19	ND	(16, 22)
δ_6	8	Pixels	(4, 12)
δ_7	50	Pixels	(46, 56)
δ_8	40	ND	(20, 60)
δ_9	5	Pixels	(3, 7)
δ_{10}	10	Pixels	(6, 14)
δ_{11}	10	Pixels	(6, 14)

*ND stand for non-dimensional.

Table III shows the heuristic values and the variation range we used for each of them. For example, the parameter δ_9 had to include at least two pixels, and as a result, its lowest limit had to be 3.

The sensitivity grade for each parameter variation is shown in Fig. 7. It refers to the accuracy rate variation achieved when only one parameter varies, while the remaining ones assume their heuristic values. We see that, depending on the parameter, the variation of the accuracy rate is in the order of magnitude 10^{-3} and can therefore be considered negligible. The results confirm that the heuristic criteria we designed, based on the Gestalt theory of perception and motor control theory, capture some properties of the writer's movements when their trajectories exhibit intersections or crossings. We can also see that δ_3 exhibits the highest sensitivity grade, meaning that it is one of the most influential parameters, and conversely, δ_6 seems to be the least influential one.

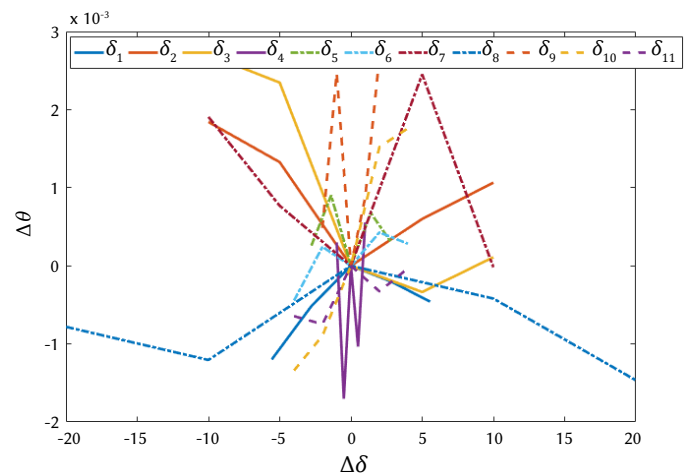


Fig. 7. Sensitivity analysis: Each line shows the variation of the accuracy rate $\Delta\theta$ for a different variation $\Delta\delta$ of the parameter δ_k .

B. Stability Study: Results

In this study, different weight values are obtained by adding Gaussian noise to the heuristic values, as: $\hat{\omega} = \mathcal{N}(\omega, (\eta \cdot \omega)^2)$ where $\eta \in (0.05, \dots, 0.5)$ is a distortion factor of the standard deviation of the Gaussian. In each experiment, the value of η was incremented by 0.05 and normalization was carried out in order to satisfy $|\omega_{ext}| + |\omega_{int}| + |\omega_{cur}| = 1$.

Table IV gives the heuristic values of the weights, whereas Fig. 8 shows the accuracy rate obtained for different distortion levels. For each value of η the figure reports the corresponding box plots. As it can be seen, the central mark, in red, is above $\theta = 0.95$ and the majority of data, in blue, are concentrated around the median. Black dots represent the outliers in each case. Consequently, highly stable performances are obtained, along with different η values. It suggested that external and internal angles, as well as the curvature conditions, are representative of the good continuity criteria.

TABLE IV. WEIGHTS FOR COMPUTING THE BRANCH POINT DIRECTIONS

Clusters	ω_{ext}	ω_{int}	ω_{cur}
Normal	0.20	0.05	0.75
T-Pattern/Retracting	0.95	0.00	0.05
Coupled	0.40	0.05	0.55
Odd-Rank	0.70	0.05	0.25

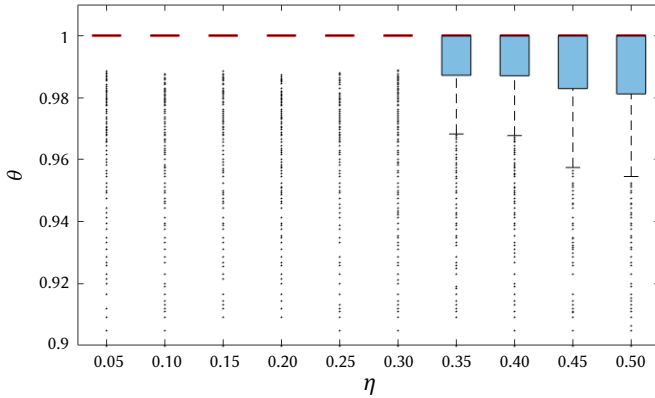


Fig. 8. Stability analysis: Vertical axis shows the accuracy rate (θ) as a function of the amount of distortion (η) applied to the heuristic values of the weights, in the horizontal axis.

V. PERFORMANCE ASSESSMENT

The overall performances of the proposed system were evaluated by comparing the recovered trajectories with the real on-line counterpart trajectories. The experiments aim to answer the following questions:

- Q1: Are the clusters correctly solved?
- Q2: Are the components correctly detected in the images?
- Q3: Does the proposed system recover the trajectories in a correct order?

We used the complete SigComp2009, SUSIG-Visual and SVC-Task2 as third-party databases for the experiments (see appendix).

A. Used Metrics

The evaluation is carried out at the pixel level between real and recovered 8-connected trajectories. To this aim, the on-line data of a specimen were interpolated to generate an 8-connected trajectory through Bresenham's line drawing algorithm [65] without any further processing [45]. As metrics, we used the Root Mean Square Error

(RMSE) [48], Signal-to-Noise-Ratio (SNR) [66] and Dynamic Time Warping (DTW) [67] to quantify the matching between them. These metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n (x_i - \hat{x}_i)^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right)} \quad (4)$$

$$\text{SNR} = 10 \log \left(\frac{\sum_{i=1}^n ((x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2)}{\sum_{i=1}^n ((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)} \right) \quad (5)$$

$$d(n, n) = \text{DTW} \left(\sum_{i=1}^n \sum_{j=1}^n \sqrt{(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2} \right) \quad (6)$$

where (x, y) and (\hat{x}, \hat{y}) are the points belonging to the real and recovered trajectories, respectively. For the sake of comparison [14], the real and estimated recovered trajectories are normalized by using a cubic spline, where its length, n , was the total sampling points in the real on-line trajectory. Next, min-max scaling is worked out.

It follows from the definition that the smaller the RMSE and DTW, the more similar the real and recovered trajectories. Conversely, higher values of the SNR represent a higher similarity between trajectories.

Furthermore, we investigate the relation between the performance and the complexity (\mathbb{C}) of the specimens, since the more complex the handwriting, the more difficult the reconstruction of its writing order. Accordingly, we define the complexity as:

$$\mathbb{C} = \alpha_1 \cdot n_c + \alpha_2 \cdot n_{r=3} + \alpha_3 \cdot n_{r>3} \quad (7)$$

where n_c denotes the number of components in the real handwriting, $n_{r=3}$ the number of 3-rank clusters and $n_{r>3}$ the number of clusters with a rank greater than three. These three factors represent the difficulty in order recovering trajectories, with the number of components being the most critical factor. Accordingly, their coefficients are empirically adjusted as: $[\alpha_1, \alpha_2, \alpha_3] = [0.6, 0.3, 0.1]$. Through the complexity formula, a histogram is obtained and divided into three parts with equal frequency binning, in a balanced fashion. To contextualize formula (7), we calculated the complexity using all specimens of the datasets used in the experiments (5972 in total). We found that 1969 samples were categorized as low complexity $\mathbb{C} \in (\alpha_1, 4.0)$; 2002 as medium complexity $\mathbb{C} \in (4.0, 6.6)$ and 2001 as high complexity $\mathbb{C} \in (6.6, 26.3)$.

B. Accuracy of Cluster Resolution (Q1)

Fig. 9b shows that our system detected 183, 877 out of 186, 165 clusters of 3 and 4-rank in all databases. These two cluster types thus represent 98.77 % of the cases handled. The accuracy rate obtained was over 97.97 % on average for the three datasets. As seen in Fig. 9a, for 3-rank clusters, the lowest and highest rates were 96.15 % for SVC-Task2 and 99.26 % for SUSIG-Visual, respectively. As mentioned in Section III, 3-rank clusters are the most challenging to solve, as they may exhibit configurations that are very similar to those of the branches attached to them. Achieving an accuracy rate greater than 96.00 % on them thus is a remarkable feat. These results show that the criteria we designed for pairing the cluster branches capture essential pieces of knowledge about human trajectory execution. They also show that the higher the cluster rank, the lower the accuracy rate. This is expected since recovering correcting higher-rank clusters is more difficult than correcting those that are lower-ranked. We also observe that there is room for improvements for branch pairings of clusters with ranks greater than 6. However, this drawback has only a limited impact on the performance because such clusters are present in less than 2.5 % of the total number of samples.

Overall, we have a global accuracy of $\theta = 98.72$ % on SigComp2009,

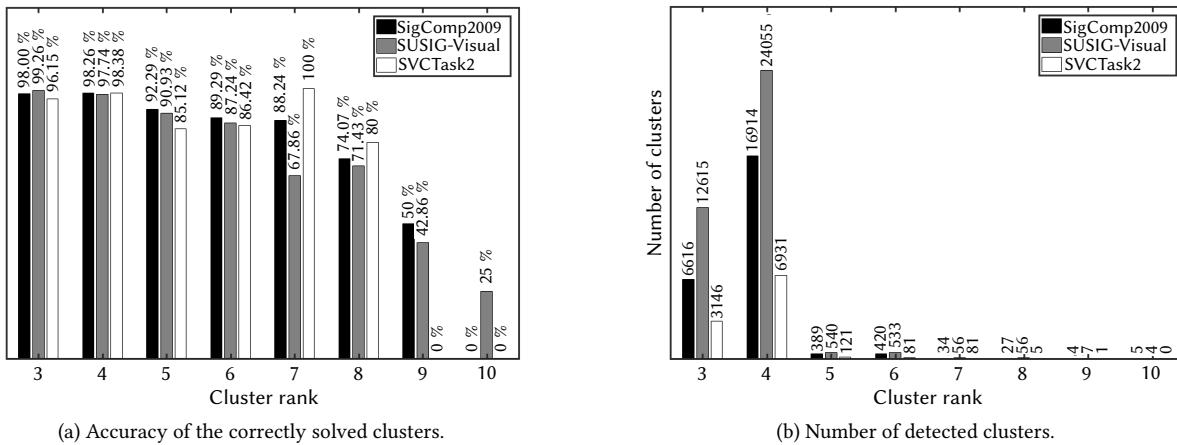


Fig. 9. Performance of the system solving the clusters found in each database.

$\theta = 98.91\%$ on SUSIG-Visual, and $\theta = 98.59\%$ on SVCTask2, which are little better than the results obtained on the SigComp2009 and SUSIG-Visual datasets [50].

C. Estimation of the Number of Components (Q2)

We assess the estimation of the number of components. Each component has two end-points, corresponding to the points where the pen-tip touches/leaves the tablet. They are used to quantify the number of components in a sample. Therefore, we compare the number of components found by our method with the actual number of components of the on-line samples.

Fig. 10 shows the density functions of the real and estimated numbers of components in the samples. We also quantify the density function similarities in terms of the Area Between Curves (ABC). The more similar the density functions, the smaller the ABC value, and therefore, the better the estimation. For all datasets, we obtain excellent performance, but in the case of the SVCTask2, it is clearly outstanding. It is explained since the other two databases contain more complicated and lengthy specimens than the SVCTask2.

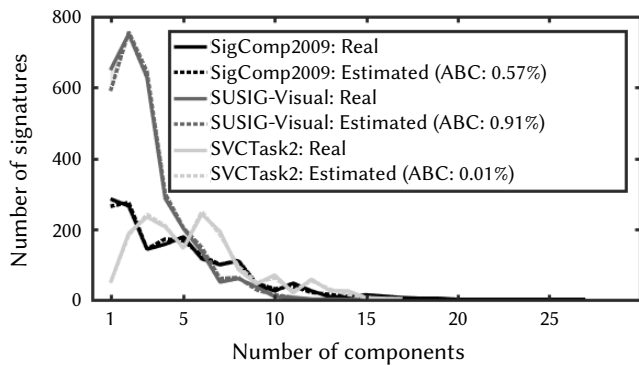


Fig. 10. Density functions of the number of components calculated with real trajectories from different databases and the estimated number of components with our system. ABC denotes the Area Between Curves.

D. Matching Between Real and Recovered Trajectories (Q3)

This experiment assesses the performance of our system when the order of a static trajectory is wholly recovered. One of the main factors influencing the complete order recovering is the selection of the starting point of each component. To evaluate the extent to which the selection of the starting points determines the recovery trajectories, we gradually relax the condition of the static samples by adding some information regarding the ending points: 1) the Estimated Starting point Nearest Criterion (ESTNC), which selects as the starting point

TABLE V. OVERALL PERFORMANCE RESULTS WHEN COMPLETE TRAJECTORIES ARE RECOVERED IN STATIC TRAJECTORIES IN TERMS OF RMSE, SNR, AND DTW

Dataset: SigComp2009		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	28.93 ± 1.26	8.15 ± 0.63	5.93 ± 0.50	14.39 ± 0.56
	RSENC	33.05 ± 1.25	10.70 ± 0.73	6.06 ± 0.55	16.65 ± 0.60
	RSEOC	41.16 ± 1.07	26.63 ± 0.72	14.63 ± 0.64	27.50 ± 0.55
RMSE	ESTNC	0.14 ± 0.01	0.27 ± 0.01	0.27 ± 0.01	0.22 ± 0.00
	RSENC	0.11 ± 0.01	0.25 ± 0.01	0.29 ± 0.01	0.22 ± 0.01
	RSEOC	0.04 ± 0.00	0.07 ± 0.01	0.16 ± 0.01	0.09 ± 0.00
DTW	ESTNC	2.73 ± 0.18	6.51 ± 0.22	8.82 ± 0.30	6.01 ± 0.15
	RSENC	2.05 ± 0.15	6.16 ± 0.23	9.39 ± 0.31	5.86 ± 0.16
	RSEOC	0.71 ± 0.09	1.82 ± 0.14	5.42 ± 0.29	2.65 ± 0.12

Dataset: SUSIG - Visual		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	23.74 ± 0.83	14.35 ± 0.64	7.75 ± 0.51	15.40 ± 0.41
	RSENC	32.32 ± 0.79	19.94 ± 0.69	9.45 ± 0.57	20.73 ± 0.44
	RSEOC	43.34 ± 0.63	32.70 ± 0.59	19.50 ± 0.62	31.98 ± 0.40
RMSE	ESTNC	0.19 ± 0.01	0.23 ± 0.01	0.28 ± 0.01	0.23 ± 0.00
	RSENC	0.11 ± 0.01	0.18 ± 0.01	0.27 ± 0.01	0.19 ± 0.00
	RSEOC	0.03 ± 0.00	0.07 ± 0.00	0.16 ± 0.01	0.08 ± 0.00
DTW	ESTNC	2.71 ± 0.10	3.70 ± 0.12	5.33 ± 0.13	3.90 ± 0.07
	RSENC	1.58 ± 0.09	3.00 ± 0.12	5.33 ± 0.14	3.28 ± 0.07
	RSEOC	0.36 ± 0.04	1.13 ± 0.08	3.05 ± 0.12	1.51 ± 0.05

Dataset: SVCTask2		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	23.72 ± 0.84	7.14 ± 0.54	2.62 ± 0.25	11.19 ± 0.41
	RSENC	31.61 ± 0.83	8.70 ± 0.65	2.11 ± 0.34	14.18 ± 0.49
	RSEOC	37.39 ± 0.63	29.30 ± 0.69	20.43 ± 0.60	29.08 ± 0.41
RMSE	ESTNC	0.13 ± 0.01	0.27 ± 0.01	0.31 ± 0.01	0.24 ± 0.00
	RSENC	0.09 ± 0.01	0.28 ± 0.01	0.35 ± 0.01	0.24 ± 0.01
	RSEOC	0.03 ± 0.00	0.06 ± 0.00	0.09 ± 0.00	0.06 ± 0.00
DTW	ESTNC	1.72 ± 0.11	3.62 ± 0.11	4.97 ± 0.10	3.43 ± 0.07
	RSENC	1.16 ± 0.10	3.74 ± 0.12	5.69 ± 0.11	3.52 ± 0.08
	RSEOC	0.34 ± 0.04	0.79 ± 0.07	1.46 ± 0.08	0.86 ± 0.04

of the next component the nearest untraced *estimated* starting point; 2) the Real Starting/Ending point Nearest Criterion (RSENC), which provides the system with the coordinates of the *real* starting and ending points of the components, and 3) the Real Starting/Ending point Ordered Criterion (RSEOC), which provides the system with the coordinates of the *real* starting and ending points, as well as the correct order of these coordinates. By comparing the results of the third and second scenarios, it is possible to estimate the efficacy of the criterion for finding the starting point of the next component. As the first scenario represents the operating condition of our method, comparing its performance with the second one allows evaluating the efficacy of the criterion for selecting the starting point of each component.

In Table V, for each database, scenario and level of complexity, we list the performance in terms of the mean and standard error of the performance measures mentioned above.

As expected, we can observe that the simpler the handwriting, the better the reconstruction, for all metrics. They also show that in case of a low complexity handwriting, the performance is very similar across the three datasets, but diverges more and more as the handwriting complexity increases.

With regards to the metrics, the performances in Table V show that the SNR is more sensitive than RMSE and DTW to the errors in recovered trajectories. Nevertheless, all metrics maintain a similar range of values in all cases.

Last but not least, the results in the table show that a significant improvement in the performance is obtained when both the starting/ending points of each component and their orders are made available to our method. This observation is independent of the database and the level of complexity.³

TABLE VI. PERFORMANCE COMPARISON OF RECOVERED TRAJECTORIES OF THE ESTIMATED STARTING POINT NEAREST CRITERION (ESTNC) WITH OTHER WORKS

Paper	Dataset	RMSE*	DTW*
– Related works about recovering words –			
[48]	Private - Single Strokes (on-line transformed in off-line)	400.98	118.12
[48]	Private - Multi Strokes (on-line transformed in offline)	503.81	171.91
[48]	Private - Scanned Words	2538.05	553.57
[48]	IRONOFF [49]	669.03	278.27
[56] (LSTM)	Unipen	-	0.04
[56] (Conv)	Unipen	-	0.04
[56] (Class Avg)	Unipen	-	0.21
– Related works about recovering signatures –			
[14] (Best system on DTW)	Public - Arabic signature [33]	0.34	28.81
[14] (Best system RMSE)	Public - Arabic signature [33]	0.25	52.40
[50]	SigComp2009	0.06	382.01
[50]	SUSIG-Visual	0.05	300.50
This work	SigComp2009	0.22	6.01
	SUSIG-Visual	0.23	3.90
	SVCTask2	0.24	3.43

* Note that different formulas for RMSE and DTW are used in the papers, making a fair comparison more difficult. We used the formulas proposed in the competition presented in [14].

³ A video showing the estimated recovered trajectories with different levels of complexity is available at <https://youtu.be/TYoZZ8CThhw>.

To put our results in context, Table VI shows the performances obtained in related works. We can see different performance ranges among the works, suggesting that no standard procedure has thus far been established for measuring the effectiveness of the writing order recovery. For this reason, beyond the implementation of RMSE and DTW metrics, it is necessary to take into account data normalization, data aggregation, and, above all, the handwriting database used, whose complexity is not easily measurable. Nonetheless, the results of such comparisons are useful, as they generally convey a rough estimate of advancements in the field, even though it does not provide a fair basis of comparison.

VI. CONCLUSIONS AND OUTLOOK

We have developed a system for recovering the ballistic trajectory order of long and complex thinned static handwritten signatures. Our system operates in three stages: (i) point classification, (ii) local examination, and (iii) global reconstruction. In the point classification, the clusters are identified and correspond to the agglomeration of pixels in the images. The agglomerations of lines correspond to crossings of lines in the thinned trajectories. Thus, a cluster can be characterized by the number of input-output lines or branches. In the local examination, input-output branches are paired by exploiting heuristic rules inspired by both good continuity and motor control principles when signing, with preference given to smooth and straight ballistic trajectories. At the global reconstruction stage, the end-points of the components (pen-downs) are identified and sorted. Once a component is re-covered, the system decides on the new component to recovering.

This procedure requires that a number of parameters and weights be adjusted. Their values are determined heuristically by trial and error, as the best matching between the real and reconstructed trajectory is sought. Furthermore, both the sensitivity and stability of the results with respect to these parameters are studied, and we see that the performance of our procedure is barely affected by a variation of up to 10 % of these parameters. To avoid overfitting in these values and make the results more meaningful, the parameters are adjusted with a subset of the SigComp2009 signature database. Then, the results are obtained with different publicly available databases, namely, the complete SigComp2009, SUSIG-Visual and SVCTask2.

The performance of the system is analyzed considering several aspects. As the ground truth of our experiments, we use the on-line trajectories, which contain details of how real signers wrote the trajectory. We first observe a competitive performance when the branches are paired on the clusters. It is worth pointing out that the branch association in the clusters is the first step towards the final writing order recovery. Moreover, a few mistakes may lead to an overall error in the estimation of trajectory order. Secondly, we also study the number of components estimated in the signature and the complete trajectory order recovery. For research purposes, our system can be freely downloaded from GitHub.

Although promising results are observed in this work, more efforts are required to ensure a more reliable estimation of the trajectories. Our experiments suggest that it is of paramount importance to continue investigating the rules for choosing the ending points of the components and their order, taking into account unknown pen-ups trajectories. Indeed, having the availability of on-line trajectories to recover a static one could improve both the cluster resolution and the complete order recovering. In this case, mapping two skeleton-based images is a further strategy that could be explored by using optical flow analysis, diffeomorphism functions, or inkball models, among others.

In a challenging framework, which uses an off-line handwriting as input and approximates its corresponding on-line counterpart as output, i.e. $(x(t), y(t))$, our system plays an important role. Nevertheless, this framework also implies that more effort is required in thinning algorithms to improve the handwriting image quality and resolution. Furthermore, estimating temporal properties in the recovered trajectories is another open question. A possible solution is to assign a timestamp sequence to the 8-connected trajectories. This would open the door to working out dynamic properties such as the velocity or acceleration. In the meantime, the proposed system constitutes a reasonable starting point for future research in the challenging field of on-line trajectory estimation from off-line specimens.

APPENDIX

A. Databases

We evaluate the proposed system on signatures because they represent long and complex handwriting, contain text and flourishes, and their patterns result in multiple pen-downs and a high number of clusters, so constituting a suitable and very challenging benchmark.

We use the following data of three publicly databases:⁴

- SigComp2009 [64]. Contains 1552 on-line Western signatures written by 79 subjects. There are 932 genuine signatures available since each signer gave 12 specimens on average.
- SUSIG-Visual corpus [68]. Includes 2820 on-line Western signatures written by 94 subjects. Each participant produced 20 genuine signatures in two sessions, i.e., $94 \times 20 = 1880$ specimens.
- SVCTask2 [69]. Consists of 1600 signatures written by 40 subjects, 17 of whom used Oriental scripts and 23, Western scripts. Each subject produced 20 genuine signatures.

SigComp2009 and SUSIG-Visual, which are made up of only Western signatures, were used in a bid to evaluate the independence of the proposed method from the database, whereas the use of SVCTask2, also composed of Oriental specimens, was intended to show the independence of the algorithm from the script type.

Eventually, the skeleton of the off-line handwriting was obtained by converting the on-line trajectories into 8-connected, one-pixel-wide digital lines through the Bresenham's line drawing algorithm [65] without any further processing [45]. The out-put resolution of the images was 600 dpi. This choice, moreover, provides a perfect spatial matching between the on-line and the off-line representations of the trajectories, establishing a solid ground truth for performance evaluation.

ACKNOWLEDGMENT

This study was funded by the Spanish government's MIMCO TEC2016-77791-C4-1-R and PID2019-109099RB-C41 research projects and European Union FEDER program/funds and by the Italian Ministry of Education, University and Research within the PRIN2015 - Handwriting Analysis against Neuromuscular Disease - HAND Project under Grant H96J16000820001.

REFERENCES

- [1] C. D. Stefano, F. Fontanella, A. Marcelli, R. Plamondon, "Graphonomics for the e-citizens: e-health, e-society and e-education," 2019. doi: 10.1016/j.patrec.2018.11.020, Graphonomics for e-citizens: e-health, e-society, e-education.
- [2] R. Plamondon, S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.
- [3] R. Plamondon, G. Pirlo, E. Anquetil, C. Rémi, H. L. Teulings, M. Nakagawa, "Personal digital bodyguards for e-security, e-learning and e-health: A prospective survey," *Pattern Recognition*, vol. 81, pp. 633–659, 2018.
- [4] C. Tomoiaga, P. Feng, M. Salzmann, P. Jayet, "Field typing for improved recognition on heterogeneous handwritten forms," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 487–493, IEEE.
- [5] S. Dash, S. K. Shakyawar, M. Sharma, S. Kaushik, "Big data in healthcare: management, analysis and future prospects," *Journal of Big Data*, vol. 6, no. 1, p. 54, 2019, doi: 10.1186/s40537-019-0217-0.
- [6] V. Rowtula, V. Bhargavan, M. Kumar, C. Jawahar, "Scaling handwritten student assessments with a document image workflow system," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 2307 – 2314.
- [7] R. Rajesh, R. Kanimozhi, "Digitized exam paper evaluation," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–5, IEEE.
- [8] I. H. Hsiao, "Mobile grading paper-based programming exams: automatic semantic partial credit assignment approach," in *European conference on technology enhanced learning*, 2016, pp. 110–123, Springer.
- [9] A. Parziale, A. Della Cioppa, R. Senatore, A. Marcelli, "A decision tree for automatic diagnosis of parkinson's disease from offline drawing samples: experiments and findings," in *International Conference on Image Analysis and Processing*, 2019, pp. 196–206, Springer.
- [10] M. Diaz, M. A. Ferrer, D. Impedovo, G. Pirlò, G. Vessio, "Dynamically enhanced static handwriting representation for parkinson's disease detection," *Pattern Recognition Letters*, vol. 128, pp. 204–210, 2019, doi: 10.1016/j.patrec.2019.08.018.
- [11] S. Colutto, P. Kahle, H. Guenter, G. Muehlberger, "Transkribus. A platform for automated text recognition and searching of historical documents," in *2019 15th International Conference on eScience (eScience)*, 2019, pp. 463–466, IEEE.
- [12] V. Nguyen, M. Blumenstein, "Techniques for static hand-writing trajectory recovery: a survey," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010, pp. 463–470, ACM.
- [13] Z. Noubigh, M. Kherallah, "A survey on handwriting recognition based on the trajectory recovery technique," in *Arabic Script Analysis and Recognition (ASAR)*, 2017 1st International Workshop on, 2017, pp. 69–73, IEEE.
- [14] A. Hassaine, S. Al Maadeed, A. Bouridane, "ICDAR 2013 competition on handwriting stroke recovery from offline data," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013.
- [15] C. De Stefano, A. Marcelli, A. Parziale, R. Senatore, "Reading cursive handwriting," in *2010 12th International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 95–100, IEEE.
- [16] P. M. Lallican, C. Viard-Gaudin, S. Knerr, "From off-line to on-line handwriting recognition," in *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 303–312. ISBN: 90-76942-01-3.
- [17] C. Viard-Gaudin, P. M. Lallican, S. Knerr, "Recognition-directed recovering of temporal information from hand-writing images," *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2537–2548, 2005, doi: 10.1016/j.patrec.2005.04.019.
- [18] B. Rabhi, A. Elbaati, H. Boubaker, A. M. Alimi, "Temporal order and pen velocity recovery for character handwriting based on sequence to sequence gated recurrent unit model," *TechRxiv*, 2020.
- [19] L. Rousseau, É. Anquetil, J. Camillerapp, "Recovery of a drawing order from off-line isolated letters dedicated to on-line recognition," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2005.
- [20] A. K. Bhunia, A. Bhowmick, A. K. Bhunia, A. Konwer, P. Banerjee, P. P. Roy, U. Pal, "Handwriting trajectory recovery using end-to-end deep encoder-decoder network," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3639–3644, IEEE.

⁴ The databases can be downloaded from the following links: Sig-Comp2009 tc11.cvc.uab.es/datasets/SigComp2009_1. SUSIG-Visual biometrics.sabanciuniv.edu/susig.html. SVCTask2 www.cse.ust.hk/svc2004/download.html.

- [21] A. Sharma, "Recovery of drawing order in handwritten digit images," in *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on*, 2013, pp. 437–441, IEEE.
- [22] A. Santoro, A. Parziale, A. Marcelli, "A human in the loop approach to historical handwritten documents transcription," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 222–227, IEEE.
- [23] B. S. Saroui, V. Sorge, "Trajectory recovery and stroke reconstruction of handwritten mathematical symbols," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, 2015, pp. 1051–1055, IEEE.
- [24] M. Diaz, M. A. Ferrer, A. Parziale, A. Marcelli, "Recovering western on-line signatures from image-based specimens," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1204–1209, IEEE.
- [25] M. Diaz, M. A. Ferrer, D. Impedovo, M. I. Malik, G. Pirlo, R. Plamondon, "A perspective analysis of handwritten signature technology," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–39, 2019, doi: 10.1145/3274658.
- [26] A. Parziale, A. Santoro, A. Marcelli, A. P. Rizzo, C. Molinari, A. G. Cappuzzo, F. Fontana, "An interactive tool for forensic handwriting examination," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 440–445, IEEE.
- [27] A. Marcelli, A. Parziale, C. De Stefano, "Quantitative evaluation of features for forensic handwriting examination," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1266–1271, IEEE.
- [28] A. Marcelli, A. Parziale, A. Santoro, "Modeling handwriting style: a preliminary investigation," in *2012 International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 411–416, IEEE.
- [29] A. Marcelli, A. Parziale, A. Santoro, "Modelling visual appearance of handwriting," in *International Conference on Image Analysis and Processing*, 2013, pp. 673–682, Springer.
- [30] C. Carmona-Duarte, M. A. Ferrer, A. Parziale, A. Marcelli, "Temporal evolution in synthetic handwriting," *Pattern Recognition*, vol. 68, pp. 233–244, 2017, doi: 10.1016/j.patcog.2017.03.019.
- [31] A. Marcelli, A. Parziale, R. Senatore, "Some observations on handwriting from a motor learning perspective," in *2nd International Workshop on Automated Forensic Handwriting Analysis*, 2013, pp. 6–10.
- [32] M. Faundez-Zanuy, J. Fierrez, M. A. Ferrer, M. Diaz, R. Tolosana, R. Plamondon, "Handwriting biometrics: Applications and future trends in e-security and e-health," *Cognitive Computation*, vol. 12, no. 5, pp. 940–953, 2020, doi: 10.1007/s12559-020-09755-z.
- [33] S. Al-Maadeed, W. Ayoubi, A. Hassaine, A. Almejali, A. Al-yazeedi, R. Al-Atiya, "Arabic signature verification dataset," in *Proceedings of the International Arab Conference on Information Technology*, 2012. ISSN: 1812-0857.
- [34] R. Plamondon, C. M. Privitera, "The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 80–91, 1999, doi: 10.1109/83.736691.
- [35] B. Kovari, "Time-efficient stroke extraction method for handwritten signatures," in *Proceedings of the 7th International Conference on Applied Computer Science - Volume 7, ACS'07*, Stevens Point, Wisconsin, USA, 2007, pp. 157–161, World Scientific and Engineering Academy and Society (WSEAS). ISBN: 9789606766183.
- [36] G. Boccignone, A. Chianese, L. P. Cordella, A. Marcelli, "Recovering dynamic information from static handwriting," *Pattern recognition*, vol. 26, no. 3, pp. 409–418, 1993, doi: 10.1016/0031-3203(93)90168-V.
- [37] S. Lee, J. C. Pan, "Offline tracing and representation of signatures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 755–771, 1992, doi: 10.1109/21.156588.
- [38] S. Lee, J. C. Pan, "Handwritten numeral recognition based on hierarchically self-organizing learning networks," in *IEEE International Joint Conference on Neural Networks*, 1991, pp. 1313–1322, IEEE.
- [39] D. S. Doermann, A. Rosenfeld, "Recovery of temporal information from static images of handwriting," *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 143–164, 1995, doi: 10.1007/BF01450853.
- [40] J. Hennebert, R. Loeffel, A. Humm, R. Ingold, "A new forgery scenario based on regaining dynamics of signature," in *International Conference on Biometrics*, 2007, pp. 366–375, Springer.
- [41] A. Elbaati, M. Kherallah, A. Ennaji, A. M. Alimi, "Temporal order recovery of the scanned handwriting," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009.
- [42] S. Jager, "Recovering writing traces in off-line handwriting recognition: using a global optimization technique," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 3, 1996, pp. 150–154 vol.3.
- [43] H. Bunke, R. Ammann, G. Kaufmann, T. M. Ha, M. Schenkel, R. Seiler, F. Eggimann, "Recovery of temporal information of cursively handwritten words for on-line recognition," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, vol. 2, 1997, pp. 931–935, IEEE.
- [44] Y. Kato, M. Yasuhara, "Recovery of drawing order from single-stroke handwriting images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 938–949, 2000, doi: 10.1109/34.877517.
- [45] Y. Qiao, M. Nishiara, M. Yasuhara, "A framework toward restoration of writing order from single-stroked handwriting image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1724–1737, 2006, doi: 10.1109/TPAMI.2006.216.
- [46] L. P. Cordella, C. De Stefano, A. Marcelli, A. Santoro, "Writing order recovery from off-line handwriting by graph traversal," in *20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 1896–1899, IEEE.
- [47] R. Senatore, A. Santoro, A. Marcelli, "From motor to trajectory plan: A feedback loop between unfolding and segmentation to improve writing order recovery," in *15th International Graphonomics Society Conference*, Cancun, Mexico, 2011, pp. 86–69. ISBN: 9780732640033.
- [48] M. Dinh, H. J. Yang, G. S. Lee, S. H. Kim, L. N. Do, "Recovery of drawing order from multi-stroke english handwritten images based on graph models and ambiguous zone analysis," *Expert Systems with Applications*, vol. 64, pp. 352–364, 2016, doi: 10.1016/j.eswa.2016.08.017.
- [49] C. Viard-Gaudin, P. M. Lallican, S. Knerr, P. Binter, "The IRESTE on/off (IRONOFF) dual handwriting database," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, 1999, pp. 455–458, IEEE.
- [50] G. Crispo, M. Diaz, A. Marcelli, M. A. Ferrer, "Tracking the ballistic trajectory in complex and long handwritten signatures," in *16th International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 351–356.
- [51] E. M. Nel, J. A. Du Preez, B. M. Herbst, "Estimating the pen trajectories of static signatures using hidden markov models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1733–1746, 2005, doi: 10.1109/TPAMI.2005.221.
- [52] K. K. Lau, P. C. Yuen, Y. Y. Tang, "Universal writing model for recovery of writing sequence of static handwriting images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 05, pp. 603–630, 2005, doi: 10.1142/S0218001405004277.
- [53] B. Zhao, M. Yang, J. Tao, "Pen tip motion prediction for handwriting drawing order recovery using deep neural network," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug 2018, pp. 704–709.
- [54] B. Zhao, M. Yang, J. Tao, "Drawing order recovery for handwriting chinese characters," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3227–3231.
- [55] B. Rabhi, A. Elbaati, Y. Hamdi, A. M. Alimi, "Handwriting recognition based on temporal order restored by the end-to-end system," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1231–1236.
- [56] T. Sumi, B. K. Iwana, H. Hayashi, S. Uchida, "Modality conversion of handwritten patterns by cross variational autoencoders," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 407–412.
- [57] R. Zhang, J. Chen, M. Yang, "Drawing order recovery based on deep learning," in *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, 2019, pp. 129–133, IEEE.
- [58] H. T. Nguyen, T. Nakamura, C. T. Nguyen, M. Nakagawa, "Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes," in *25th International Conference on Pattern Recognition (ICPR)*, 2020, IEEE.
- [59] C. De Stefano, M. Garruto, A. Marcelli, "A saliency-based multiscale

method for on-line cursive handwriting shape description,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 6, pp. 1139–1156, 2004. ISSN:0218-0014.

- [60] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.
- [61] F. H. Allport, *Theories of perception and the concept of structure: A review and critical analysis with an introduction to a dynamic-structural theory of behavior*. John Wiley & Sons Inc, 1955.
- [62] R. Plamondon, “A kinematic theory of rapid human movements: Part III. kinetic outcomes,” *Biological Cybernetics*, vol. 78, no. 2, pp. 133–145, 1998, doi: 10.1007/s004220050420.
- [63] T. Steinherz, D. Doermann, E. Rivlin, N. Intrator, “Offline loop investigation for handwriting analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 193–209, 2008, doi: 10.1109/TPAMI.2008.68.
- [64] V. L. Blankers, C. E. van den Heuvel, K. Y. Franke, L. G. Vuurpijl, “ICDAR 2009 signature verification competition,” in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 1403–1407, IEEE.
- [65] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965, doi: 10.1147/sj.41.0025.
- [66] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, R. Plamondon, “idelog: Iterative dual spatial and kinematic extraction of signalognormal parameters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 114–125, 2020, doi: 10.1109/TPAMI.2018.2879312.
- [67] M. Diaz, P. Henriquez, M. A. Ferrer, G. Pirlo, J. B. Alonso, C. Carmona-Duarte, D. Impedovo, “Stability-based system for bearing fault early detection,” *Expert Systems with Applications*, vol. 79, pp. 65–75, 2017, doi: 10.1016/j.eswa.2017.02.030.
- [68] A. Kholmatov, B. Yanikoglu, “SUSIG: an on-line signature database, associated protocols and benchmark results,” *Pattern Analysis and Applications*, vol. 12, no. 3, pp. 227–236, 2009, doi: 10.1007/s10044-008-0118-x.
- [69] D. Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, G. Rigoll, “SVC2004: First international signature verification competition,” in *International conference on biometric authentication*, 2004, pp. 16–22, Springer.

Moises Diaz



Moises Diaz received an M.Tech., an M.Sc., and a Ph.D. in Engineering, and an M.Ed. degree in secondary education from La Universidad de Las Palmas de Gran Canaria, Las Palmas, Spain, in 2010, 2011, 2016, and 2013, respectively. He is currently an associate professor at Universidad del Atlántico Medio, Spain. His current research interests include pattern recognition, document analysis, handwriting recognition, biometrics, computer vision, and intelligent transportation systems.

Gioele Crispo



Gioele Crispo is a Machine Learning Engineer focused on NLP and deep learning. AWS Machine Learning certified. He received a M.Sc. in Computer Engineering from the University of Salerno (Italy). He collaborates with the University of Salerno, the University of the Middle Atlantic, and the University of Las Palmas de Gran Canaria (ULPGC) for research in machine learning, pattern recognition, and handwriting. His main interests concern deep learning and the application of transformers and attention mechanisms to images and music.

Antonio Parziale



Antonio Parziale received a Master’s degree in Electronic Engineering and a Ph.D. in Information Engineering from the University of Salerno, Italy, in 2009 and 2016, respectively. He does his research at the Natural Computation Laboratory of the University of Salerno since 2010. In 2012, he has co-founded a spin-off company devoted to providing software technologies for automatic document processing. His research interests include handwriting analysis for early diagnosis of neurodegenerative diseases, signature and writer verification, systems neuroscience, and motor control.

Angelo Marcelli



Angelo Marcelli is a professor of computer engineering at the Department of Electrical and Information Engineering and Applied Mathematics of the University of Salerno (Italy), where he is the founder and head of the Natural Computation lab. A. Marcelli is Associate Editor of Pattern Recognition Letters and Associate Editor-in-chief of the Journal of Forensic Document Examination and served as President of the International Graphonomics Society. His research focuses on handwriting, where he has contributed to developing methods and tools for handwriting analysis and recognition and their application to forensic handwriting analysis, historical document processing, on-line and off-line handwriting recognition. More recently, he has been investigating the human motor system and developed neurocomputational models to elicit the role that different areas of the brain and the spinal cord play in hand-writing learning and generation and the way neurodegenerative diseases affect handwriting production. He has authored more than 150 papers in journals, books, and conference proceedings and holds an international patent on these topics.

Miguel A. Ferrer



Miguel A. Ferrer received an M.Sc. and a Ph.D. from the Universidad Politécnica de Madrid, Madrid, Spain, in 1988 and 1994, respectively. He joined the University of Las Palmas de Gran Canaria, Las Palmas, Spain, in 1989, where he is currently a Full Professor. He established the Digital Signal Processing Research Group in 1990. His current research interests include pattern recognition, biometrics, audio quality, and computer vision applications to fisheries and aquaculture..