

8-2018

# Offline Signature Verification via Structural Methods: Graph Edit Distance and Inkball Models

Paul Maergner

*University of Fribourg, Switzerland*

Nicholas Howe

*Smith College, [nhowe@smith.edu](mailto:nhowe@smith.edu)*

Kaspar Riesen

*University of Applied Sciences and Arts Western Switzerland*

Rolf Ingold

*University of Fribourg, Switzerland*

Andreas Fischer

*University of Fribourg, Switzerland*Follow this and additional works at: [https://scholarworks.smith.edu/csc\\_facpubs](https://scholarworks.smith.edu/csc_facpubs)Part of the [Computer Sciences Commons](#)

## Recommended Citation

Maergner, Paul; Howe, Nicholas; Riesen, Kaspar; Ingold, Rolf; and Fischer, Andreas, "Offline Signature Verification via Structural Methods: Graph Edit Distance and Inkball Models" (2018). Computer Science: Faculty Publications, Smith College, Northampton, MA.

[https://scholarworks.smith.edu/csc\\_facpubs/136](https://scholarworks.smith.edu/csc_facpubs/136)

This Conference Proceeding has been accepted for inclusion in Computer Science: Faculty Publications by an authorized administrator of Smith ScholarWorks. For more information, please contact [scholarworks@smith.edu](mailto:scholarworks@smith.edu)

# Offline Signature Verification via Structural Methods: Graph Edit Distance and Inkball Models

Paul Maergner\*, Nicholas R. Howe<sup>§</sup>, Kaspar Riesen<sup>‡</sup>, Rolf Ingold\* and Andreas Fischer\*<sup>†</sup>

\*Department of Informatics, University of Fribourg, Fribourg, Switzerland

<sup>†</sup>Institute of Complex Systems, University of Applied Sciences and Arts Western Switzerland, Fribourg, Switzerland

<sup>‡</sup>Institute for Information Systems, University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland

<sup>§</sup>Department of Computer Science, Smith College, Northampton, Massachusetts, USA

paul.maergner@unifr.ch, nhowe@cs.smith.edu, kaspar.riesen@fhnw.ch, rolf.ingold@unifr.ch, andreas.fischer@unifr.ch

**Abstract**—For handwritten signature verification, signature images are typically represented with fixed-sized feature vectors capturing local and global properties of the handwriting. Graph-based representations offer a promising alternative, as they are flexible in size and model the global structure of the handwriting. However, they are only rarely used for signature verification, which may be due to the high computational complexity involved when matching two graphs. In this paper, we take a closer look at two recently presented structural methods for handwriting analysis, for which efficient matching methods are available: keypoint graphs with approximate graph edit distance and inkball models. Inkball models, in particular, have never been used for signature verification before. We investigate both approaches individually and propose a combined verification system, which demonstrates an excellent performance on the MCYT and GPDS benchmark data sets when compared with the state of the art.

**Index Terms**—offline signature verification, structural pattern recognition, graph edit distance, inkball models

## I. INTRODUCTION

Handwritten signatures are broadly used for personal authentication and there has always been an interest in verifying their authenticity. Unfortunately, the verification of signatures often has to rely on only a few genuine specimens. This makes signature verification a challenging task even for humans. Nevertheless, it has been shown that state-of-the-art automatic signature verification systems are able to achieve a level of accuracy that is similar to other biometric systems [1].

The development of automatic signature verification systems remains an active field of research. Hereby, the pattern recognition community distinguishes two different cases of signature verification: *online* signature verification uses dynamic characteristics, like timing information, speed, and pressure, while *offline* signature verification is limited to static information, i.e. the image of the signature. The offline case applies to more use cases, but it is also the more difficult task [2]. In this work, we consider the offline case.

Commonly, state-of-the-art systems for offline signature verification employ *statistical pattern recognition*, i.e. they represent the handwriting with fixed-size feature vectors. These vectors consist of either local information, such as histogram of oriented gradients (HOG), local binary patterns (LBP), or Gaussian grid features taken from signature contours [3], or

global information, e.g. geometrical features like number of branches in the skeleton, Fourier descriptors, number of holes, moments, projections, distributions, position of barycenter, tortuosities, directions, curvatures and chain codes [1], [4]. For signature verification, these feature vectors are then used in conjunction with statistical classifiers, such as dynamic time warping (DTW), support vector machines (SVM), or hidden Markov models (HMM) [5].

A more powerful representation formalism is offered by *graphs* used for *structural pattern recognition*. Graphs consist of nodes and edges, which model relations between the nodes. For signatures, these nodes commonly represent keypoints on the signature or elementary strokes. Relations that exist between these parts in the global structure of the signature are modeled with edges. However, the power of graphs comes at the expense of high computational complexity [6], which may be the reason why they have been only rarely used for signature verification so far. Examples include the early proposal to represent signatures based on stroke primitives by Sabourin et al. [7], the modular graph matching approach proposed by Bansal et al. [8], and the use of basic concepts of graph theory by Fotak et al. [9].

Recently, Maergner et al. [10] have introduced a general framework for graph-based signature verification based on the graph edit distance between labeled graphs. Promising verification results are reported for so-called keypoint graphs that have also been used for handwriting recognition [11] and keyword spotting [12] before. To overcome the high computational complexity of matching two graphs, they use a bipartite approximation of the graph edit distance [13].

Another promising approach to structural handwriting analysis are inkball models. They have been introduced by Howe in [14] as a technique for performing segmentation-free word spotting when limited training data are available. Later, they have also been used as a complex feature with HMM for handwriting recognition [15]. Inkball models share some similar properties with keypoint graphs. However, inkball models are rooted trees that are directly and efficiently matched with a skeleton image.

In this paper, we investigate inkball models for the first time for signature verification. Furthermore, we propose a

combined system that integrates keypoint graphs and inkball models as two complementary handwriting models. On two benchmark data sets, we evaluate the two structural methods individually, combined, and in comparison with the current state of the art.

In the remainder of this paper, we formally introduce keypoint graphs and graph edit distance in Section II, and inkball models in Section III. Then, we elaborate on how we use them individually as well as combined for offline signature verification in Section IV. Afterwards, we evaluate the different approaches on two publicly available data sets and compare our results with the state of the art Section V. Finally, we present our conclusion and outlook in Section VI.

## II. GRAPH EDIT DISTANCE

The first structural method considered in this paper is the approach introduced by Maergner et al. in [10]. In their approach signature images are binarized and thinned into skeleton images and from that keypoint graphs are created. These graphs are then compared using an approximation of the graph edit distance. Afterwards the resulting graph edit cost is normalized. These steps are briefly reviewed in the next subsections. For more details, we refer the reader to [10].

### A. Image Processing

The signature images are binarized and skeletonized. First, a difference of Gaussians filter is applied on the grayscale image. Afterwards, a global threshold is used to create a binary image. Lastly, the thinning algorithm introduced by Zhang and Suen in [16] is applied to get a skeleton image.

### B. Graph Representation

Formally, a labeled graph is defined as a four-tuple  $g = (V, E, \mu, \nu)$ , where

- $V$  is the finite set of nodes,
- $E \subseteq V \times V$  is the set of edges,
- $\mu : V \rightarrow L_V$  is the node labeling function,
- $\nu : E \rightarrow L_E$  is the edge labeling function.

Keypoint graphs are extracted from a skeleton image of handwriting (for an example, see Fig. 1). Nodes represent points on the skeleton and they are labeled with the coordinates of these points. The edges are unlabeled and undirected and they connect nodes that are next to each other on the skeleton.

The points that are represented by nodes are the end- and junction-points of the skeleton. Then, the left outer most pixel of circular structures are added if they do not contain any end- or junction-points. Afterwards, additional points are added by tracing along the skeleton and adding points after traveling a distance of  $D_{\text{keypoint}}$  without hitting an already selected point.

After the graph has been created, the node labels are centered so that the average of all node labels is equal to  $(0, 0)$ . This normalization ensures that the graph representation is translation-invariant.

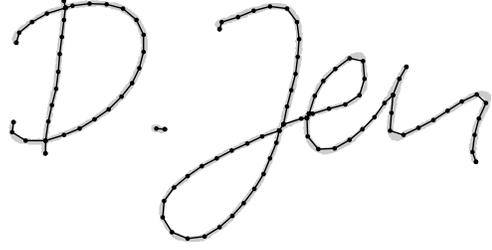


Fig. 1. Keypoint Graph

### C. Approximated Graph Edit Distance

One of the most flexible ways to compare two graphs is *graph edit distance* (GED) [17], [18]. It measures the dissimilarity of two graphs by calculating the cost of the cheapest transformation of graph  $g_1 = (V_1, E_1, \mu_1, \nu_1)$  into graph  $g_2 = (V_2, E_2, \mu_2, \nu_2)$ . Hereby, a transformation consists of a sequence of edit operations, which are commonly defined as substitutions, deletions, and insertions of nodes and edges respectively. Given an appropriate cost function, graph edit distance can handle any kind of labeled graph. Unfortunately, the computation time of the exact GED is exponential in the number of nodes of the two graphs. Therefore, exact GED is in practice only applicable to rather small graphs.

To overcome the computational problem, a bipartite approximation of GED proposed by Riesen and Bunke [13] is used. Their approximation framework reduces the computation of GED to an instance of a *linear sum assignment problem* (LSAP) with cubic complexity. The lower bound of GED introduced in [19] is considered.

### D. Cost Function

The node substitution cost is the Euclidean distance between the node labels. Formally,

$$c(u \rightarrow v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2},$$

where  $(x_u, y_u)$  and  $(x_v, y_v)$  are the labels (i.e. coordinates) of nodes  $u$  and  $v$  respectively.

The insertion and deletion costs of nodes and edges rely on the average length  $m(g_1)$  of all edges in the graph  $g_1$ . Formally, the node deletion and insertion cost is defined as

$$c(u \rightarrow \varepsilon) = c(\varepsilon \rightarrow v) = m(g_1),$$

and the edge deletion and insertion cost as

$$c(e_1 \rightarrow \varepsilon) = c(\varepsilon \rightarrow e_2) = 2 \cdot m(g_1).$$

The edge substitution cost is set to zero:  $c(e_1 \rightarrow e_2) = 0$ .

### E. Normalization of Graph Edit Distance

The graph edit distance is normalized with the maximal graph edit distance possible when comparing the two graphs. The maximal graph edit distance is the cost of deleting all nodes and edges from the first graph and inserting all the nodes and edges from the second graph. This normalization

calculates how close the dissimilarity is to the maximal dissimilarity instead of just the distance. Formally, the graph edit distance based comparison of two signature images  $r$  and  $t$  is defined as follows:

$$d_{\text{GED}}(r, t) = \frac{\text{GED}(g_r, g_t)}{\text{GED}_{\max}(g_r, g_t)}, \quad (1)$$

where  $g_r$  and  $g_t$  are the keypoint graphs of the signatures images  $r$  and  $t$  respectively,  $\text{GED}(g_r, g_t)$  is the lower bound of the graph edit distance calculated using the bipartite graph matching framework, and  $\text{GED}_{\max}(g_r, g_t)$  is the maximal graph edit distance.

### III. INKBALL MODELS

The second structural method considered in this work is the inkball model introduced by Howe in [14]. An inkball model can be generated from a signature image through a procedure similar to that used for keypoint graphs. After binarization and thinning [20], inkballs are placed at each junction and endpoint. Additional inkball nodes are added sequentially, as close as possible to the existing positions while staying at least distance  $\sqrt{2} \cdot D_{\text{inkball}}$  away. When no more can be added in this manner, additional inkballs are added that are as close as possible to existing positions but at least distance  $D_{\text{inkball}}$  away.

Once a set of inkballs have been identified, they are linked in a tree structure by greedily adding edges between the closest nodes that are not yet connected. The node nearest to the center of mass is arbitrarily designated as the root, and each child node is annotated with the Cartesian offset (denoted  $\vec{o}_i$ ) of its parent node relative to its own position.

#### A. Inkball Matching Energy

The degree of fit between an inkball model derived from signature sample  $r$  to a second signature sample  $t$  can be measured by a combination of deformation and proximity: how closely the inkballs can be placed near the observed ink, and how much the model structure must be deformed to achieve that proximity (for a matching example, see Fig. 2). Formally, define a *configuration*  $C$  of the model as a placement  $\vec{v}_i$  for each inkball. These imply *configuration offsets*  $\vec{s}_i = \vec{v}_{i\uparrow} - \vec{v}_i$  for all nodes except the root. The configuration energy then becomes the sum of squared differences between the configuration offsets and the original the model offsets, plus the square of the minimum distances  $\Omega_t(\vec{v}_i)$  from each inkball location to the target skeleton.

$$E(r, C, t) = E_{\xi}(r, C) + \lambda E_{\Omega}(C, t) \quad (2)$$

$$E_{\xi}(r, C) = \sum_{i=2}^n \|\vec{s}_i - \vec{o}_i\|^2 \quad (3)$$

$$E_{\Omega}(C, t) = \sum_{i=1}^n \Omega_t(\vec{v}_i)^2 \quad (4)$$

In practice any mismatch larger than a certain threshold should be treated as equally bad, and thus a truncated quadratic

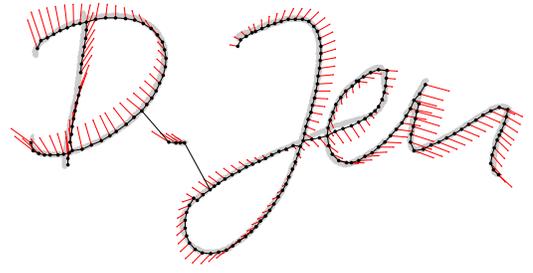


Fig. 2. Inkball Model Matching

will better serve for the energy function. The truncated energy is computed in a structure-aware fashion:

$$E'_i(r, C, t) = \min(E_i(r, C, t), N_i \tau) \quad (5)$$

$$E_i(r, C, t) = \Omega_T(\vec{v}_i)^2 + \sum_{j \in i\downarrow} [\|\vec{s}_j - \vec{o}_j\|^2 + E'_j(r, C, t)] \quad (6)$$

Here  $\tau$  represents a maximal per-node energy contribution,  $i\downarrow$  denotes the children of node  $i$ , and  $N_i$  is the number of inkball nodes in the subtree with root at node  $i$ . For purposes of signature verification, the important quantity is the minimal configuration energy, which can be efficiently computed using dynamic programming (for further algorithmic details, see Howe et al. [15]).

$$E^*(r, t) \equiv \min_C E'_1(r, C, t) \quad (7)$$

In the experiments,  $D_{\text{inkball}}$  and  $\tau$  are free parameters whose values can be optimized on a validation set.

#### B. Normalization of the Inkball Deformation Energy

Using the equations above, two signature images may be compared by converting the first one to an inkball model and computing the optimal match  $E^*$  to the other image. The score obtained is highly influenced by the number of inkballs in the model, and thus is not independent of image scale. To address this shortcoming it is preferable to apply a normalization similar to that used with the graph edit distance. Instead of measuring the amount of energy needed to match the inkball model to a signature, we measure how much energy is needed on average per inkball. Formally, we define the inkball based dissimilarity of two signature images  $r$  and  $t$  as follows:

$$d_{\text{inkball}}(r, t) = \frac{E^*(r, t)}{N_r} \quad (8)$$

### IV. SIGNATURE VERIFICATION SYSTEM

In the task of offline signature verification, an unseen signature image claiming to be from a specific user is compared with known signatures from that user, so-called *references*. Based on these comparisons a dissimilarity score between the reference signatures and the questioned signature is calculated. If this score is below a certain threshold, the signature is accepted, otherwise it is rejected.

### A. Reference-based Normalization

Each verification score (either  $d_{\text{GED}}$  or  $d_{\text{inkball}}$ ) is divided by the average dissimilarity score of the reference signatures of the current user as suggested in [10]. Formally,

$$\hat{d}(r, t) = \frac{d(r, t)}{\delta(R)}, \quad (9)$$

where  $t$  is a questioned signature image,  $r \in R$  is a reference signature image,  $R$  is the set of all reference signature images of the current users, and

$$\delta(R) = \frac{1}{|R|} \sum_{r \in R} \min_{s \in R \setminus r} d(r, s).$$

### B. Signature Verification Score

We consider the minimum dissimilarity over all reference signatures  $R$  of the claimed user for accepting or rejecting the questioned signature  $t$ . Formally,

$$d(R, t) = \min_{r \in R} \hat{d}(r, t) \quad (10)$$

### C. Multiple Classifier System

Additionally to the signature verification score based on a single dissimilarity measure, we propose a multiple classifier system (MCS) using a linear combination of the two dissimilarity measures as our combined dissimilarity score. Before the linear combination, each dissimilarity score is z-score normalized based on all references signature images in the current data set.

$$d_{\text{MCS}, \alpha}(R, t) = \min_{r \in R} \left( \alpha \cdot \hat{d}_{\text{GED}}^*(r, t) + (1 - \alpha) \cdot \hat{d}_{\text{inkball}}^*(r, t) \right), \quad (11)$$

where  $\alpha \in [0, 1]$ , and

$$\hat{d}^*(r, t) = \frac{\hat{d}(r, t) - \mu_{\mathcal{R}}}{\sigma_{\mathcal{R}}}, \quad (12)$$

considering the mean  $\mu_{\mathcal{R}}$  and the standard deviation  $\sigma_{\mathcal{R}}$  calculated over the set  $\mathcal{R} = \{R_1, \dots, R_n\}$  of all references signature sets  $R_i$  of all  $n$  users in the current data set. For example, the mean  $\mu_{\mathcal{R}}$  is calculated as

$$\mu_{\mathcal{R}} = \frac{1}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} \left( \frac{1}{|R|} \sum_{r \in R} \min_{s \in R \setminus r} \hat{d}(r, s) \right), \quad (13)$$

## V. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the two structural methods, individually as well as in combination, on two publicly available benchmark data sets. The focus lies on distinguishing genuine signatures from *skilled forgeries (SF)*, which are forgeries that have been created for each user with knowledge about the user's genuine signatures. Additionally, we test how well genuine signatures can be distinguished from *random forgeries (RF)*, which are signatures of other users that are used for a brute force attack on the verification system. The performance on both is measured using the *equal error rate (EER)*. The EER is the point where the false rejection rate is equal to the false acceptance rate in the *detection error tradeoff (DET)* curve.

### A. Data Sets

For our experimental evaluation, we consider the following publicly available signature image data sets.

*GPDSsynthetic-Offline*: Ferrer et al. have introduced this data set in [21]. It contains 24 genuine signatures and 30 simulated forgeries for each of the 4,000 synthetic users. We have created two subsets of this data set using the first  $n$  users ( $n \in \{10, 75\}$ ). The subsets are called *GPDS-10* and *GPDS-75* respectively. Hence, GPDS-10 contains  $10 \cdot (30 + 24) = 540$  signature images and GPDS-75 contains  $75 \cdot (30 + 24) = 4,050$  signature images.

*MCYT-75*: Ortega-Garcia et al. introduced this data set as part of the MCYT baseline corpus in [22], [23]. For each of the 75 users, the data set offers 15 genuine signatures and 15 skilled forgeries. Thus,  $75 \cdot (15 + 15) = 2,250$  signature images are included in MCYT-75. All available genuine signatures and skilled forgeries are used in our experiments.

### B. Tasks

We evaluate the following commonly used tasks:

- R5: First five genuine signatures are used as references.
- R10: First ten genuine signatures are used as references.

The remaining genuine signatures are used for testing for both the skilled forgery (SF) and the random forgery (RF) evaluation. All skilled forgeries are used for the SF evaluation. For the RF evaluation, we used the first genuine signature of all other users as random forgeries. This means that for example for MCYT-75 R10, we have  $75 \cdot 10 = 750$  reference signatures,  $75 \times 5 = 375$  genuine signatures,  $75 \times 15 = 1,125$  skilled forgeries, and  $75 \times 74 = 5,550$  random forgeries.

### C. Setup

We performed a grid search on GPDS-10 R10 SF to optimize the parameters of the inkball dissimilarity. The grid search was performed over the following parameter range:  $D_{\text{inkball}} \in \{2, 4, 8, 16, 32\}$  and  $\tau \in \{16, 32, 64, 128, \infty\}$ . Additionally, we tested all of these configurations with and without inkball normalization (see Section III-B). The best results have been achieved using  $s = 4$ ,  $\tau = 32$ , and with normalization turned on. Note that decreasing the  $D_{\text{inkball}}$  increases the size of the model and therefore the calculation time.

For the graph edit distance based dissimilarity, we use the configuration proposed in [10] where they optimized the parameters on GPDS-75. Specifically, we used  $D_{\text{keypoint}} = 25$ .

In an additional validation step on GPDS-10 R10, we tested different weights for our proposed multiple classifier system (see Section IV-C). The best results have been achieved using  $\alpha = 0.4$  when investigating  $\alpha \in \{0, 0.1, 0.2, \dots, 1\}$ .

### D. Results on MCYT-75 and GPDS-75

Our EER results are shown in Table I for SF and in Table II for RF. DET curves for the SF experiment are shown in Fig.3. Consistently, the inkball dissimilarity performs better with normalization than without. Interestingly, the performance of the two dissimilarities differs depending on the data set. While

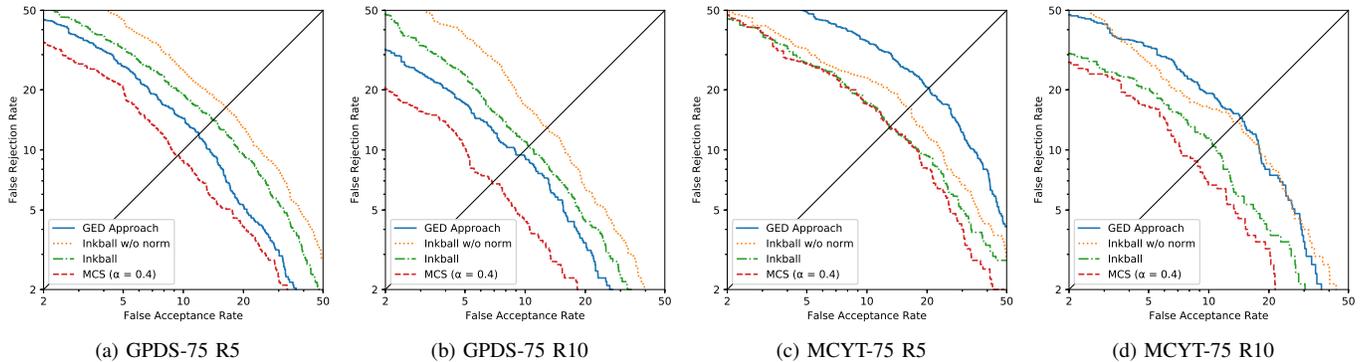


Fig. 3. DET curves.

TABLE I  
EER ON GPDS-75/MCYT-75 (SF). RESULTS ON SKILLED FORGERIES (SF) USING THE FIRST 5 OR 10 GENUINE AS REFERENCES.

System	GPDS-75		MCYT-75	
	R5	R10	R5	R10
Maergner et al. [10] GED app.	11.96	9.42	20.36	14.40
Proposed inkball	16.31	12.84	16.09	13.87
Proposed inkball normalized	14.09	10.36	12.98	10.49
Proposed MCS ( $\alpha = 0.4$ )	9.42	6.84	13.07	8.71

TABLE II  
EER ON GPDS-75/MCYT-75 (RF). RESULTS ON RANDOM FORGERIES (RF) USING THE FIRST 5 OR 10 GENUINE AS REFERENCES.

System	GPDS-75		MCYT-75	
	R5	R10	R5	R10
Maergner et al. [10] GED app.	4.90	3.60	6.25	2.92
Proposed inkball	10.22	7.21	5.86	4.52
Proposed inkball normalized	7.75	5.51	5.19	3.46
Proposed MCS ( $\alpha = 0.4$ )	3.66	2.05	3.06	1.24

the GED approach on its own is better on GPDS-75, the inkball approach is better on MCYT-75. However, on both data sets the lowest EER is achieved by using the proposed MCS combination of GED and inkball. Only in the case of MCYT-75 R5 SF, the EER is minimally higher. But the DET curve in Fig. 3c shows that the MCS performs better overall.

#### E. Comparison with State-of-the-Art

Various evaluation protocols have been used in different publications about offline signature verification. We have identified a group of publications that have used the same data sets. Then, we have followed their evaluation protocol and have compared our results with the reported results.

1) *Comparison on GPDS-75 and MCYT-75*: Not many results have been published that use the relatively new GPDSsynthetic-Offline data set. We compare our results against the results published in [10] and the results presented on the GPDS website<sup>1</sup>. The results from the GPDS website have been obtained by leveraging a system that was published in [24]. The following protocol has been used:

<sup>1</sup><http://www.gpds.ulpgc.es/downloadnew/download.htm> (March 24, 2018)

TABLE III  
COMPARISON ON GPDS-75/MCYT-75. AVERAGE EER RESULTS OVER 10 RANDOM SELECTIONS OF REFERENCE SIGNATURES. EVALUATED SF AND RF FROM GPDS-75 AND MCYT-75.

System	GPDS-75 R10		MCYT-75 R10	
	RF	SF	RF	SF
Ferrer et al. [24] <sup>1</sup>	0.76*	16.01	0.35*	11.54
Maergner et al. [10] GED app.	2.73	8.29	2.83	12.01
Proposed inkball	7.17	12.92	4.17	10.87
Proposed inkball normalized	5.22	10.64	3.13	8.29
Proposed MCS ( $\alpha = 0.4$ )	1.99	6.67	1.88	7.20

\*: All genuine signatures of other users as RF

Select 10 reference signatures per user randomly and average the results over 10 runs using different references. Table III shows our results using the same protocol compared with the previously published results. On SF, the combination of GED and inkball achieves the lowest EER, while Ferrer et al. obtain the lowest EER on RF.

2) *Comparison on MCYT-75*: Four publications on the MCYT-75 data set have published results using the a posteriori user-dependent score normalization presented in [25]. This normalization ignores the issue of user adaptation by calculating the EER individually for each user and then subtracting the corresponding threshold value from the user's scores. Table IV shows the results on MCYT-75 R5 and R10 of the four publications as well as our results using the same a posteriori score normalization. The lowest EER has been achieved using our proposed MCS system.

## VI. CONCLUSIONS AND OUTLOOK

The two structural methods investigated in this paper have demonstrated an excellent signature verification performance on the MCYT and GPDS benchmark data sets, especially when combined. To our knowledge, this is the first time that inkball models have been used for the task of signature verification. We have shown that a normalization of the inkball matching score leads to a reduction of the equal error rate (EER). Compared to a structural approach based on graph edit distance, the inkball models achieved a lower EER on MCYT, but a higher EER on GPDS. However, combining both structural approaches in a multiple classifier system improved

TABLE IV

COMPARISON ON MCYT-75 R5/R10. EER RESULTS FOR SKILLED FORGERIES WITH *a posteriori* USER-DEPENDENT SCORE NORMALIZATION. THE FIRST 5 OR 10 GENUINE SIGNATURES ARE USED AS REFERENCES FOR R5 AND R10 RESPECTIVELY.

System	MCYT-75 R5		MCYT-75 R10	
	RF	SF	RF	SF
Alonso-Fernandez et al. [26]	9.79*	23.78	7.26*	22.13
Fierrez-Aguilar et al. [23]	2.69**	11.00	1.14**	9.28
Gilperez et al. [27]	2.18*	10.18	1.18*	6.44
Maergner et al. [10] GED app.	2.40	14.49	1.89	11.64
Proposed inkball	3.10	10.31	2.07	10.76
Proposed inkball normalized	2.88	9.33	2.02	8.53
Proposed MCS ( $\alpha = 0.4$ )	0.92	9.07	0.52	5.78

\*: All genuine signatures of other users as RF

\*\* : First 5 genuine signatures from each other user as RF.

the performance of our system on both data sets. This suggests that both structural approaches look at complementary aspects of the signature images. The multiple classifier system using inkball models as well as graph edit distance has achieved the lowest EER compared to previously published results on skilled forgeries using the exact same evaluation protocol.

We see several future lines of research related to structural pattern recognition. First, we want to extend the matching process by including a signature stability measure to improve the distinction between genuine signatures and forgeries. Modeling the stability for each user individually is a challenging task, but we believe that the structural representation offers promising ways to tackle this. Additionally, we would like to combine the proposed structural verification systems with statistical verification systems. We think the complementary perspectives on the signature images is likely to improve the robustness of biometric authentication even further.

#### ACKNOWLEDGMENT

This work has been supported by the Swiss National Science Foundation project 200021\_162852.

#### REFERENCES

- [1] Donato Impedovo and Giuseppe Pirlo. Automatic signature verification: The state of the art. *IEEE Trans. on Systems, Man and Cybernetics Part C: Applications and Reviews*, 38(5):609–635, 2008.
- [2] Rejean Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
- [3] Mustafa Berkay Yilmaz, Berrin Yanikoglu, Caglar Tirkaz, and Alisher Kholmatov. Offline signature verification using classifier combination of HOG and LBP features. In *Proc. Int. Joint Conference on Biometrics*, pages 1–7, 2011.
- [4] Réjean Plamondon and Guy Lorette. Automatic signature verification and writer identification - the state of the art. *Pattern Recognition*, 22(2):107–131, 1989.
- [5] M.A. Ferrer, J.B. Alonso, and C.M. Travieso. Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(6):993–997, 2005.
- [6] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.
- [7] R. Sabourin, R. Plamondon, and L. Beaumier. Structural interpretation of handwritten signature images. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 8(3):709–748, 1994.
- [8] Abhay Bansal, Bharat Gupta, Gaurav Khandelwal, and Shampa Chakraverty. Offline signature verification using critical region matching. *Int. Journal of Signal Processing, Image Processing and Pattern*, 2(1):57–70, 2009.
- [9] Tomislav Fotak, Miroslav Baca, and Petra Koruga. Handwritten signature identification using basic concepts of graph theory. *WSEAS Transactions on Signal Processing*, 7(4):145–157, 2011.
- [10] Paul Maergner, Kaspar Riesen, Rolf Ingold, and Andreas Fischer. A structural approach to offline signature verification using graph edit distance. In *Proc. of International Conference on Document Analysis and Recognition (ICDAR)*, pages 1216–1222. IEEE, 2017.
- [11] Andreas Fischer, Kaspar Riesen, and Horst Bunke. Graph similarity features for HMM-based handwriting recognition in historical documents. In *Proc. 12th Int. Conf. on Frontiers in Handwriting Recognition*, pages 253–258, 2010.
- [12] Michael Stauffer, Andreas Fischer, and Kaspar Riesen. Graph-based keyword spotting in historical handwritten documents. In *Proc. Int. Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 564–573, 2016.
- [13] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 6 2009.
- [14] N. Howe. Part-structured inkball models for one-shot handwritten word spotting. In *Proc. of International Conference on Document Analysis and Recognition (ICDAR)*, 2013.
- [15] N. Howe, A. Fischer, and B. Wicht. Inkball models as features for handwriting recognition. In *Proc. of International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016.
- [16] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.
- [17] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 5 1983.
- [18] Kaspar Riesen. *Structural Pattern Recognition with Graph Edit Distance*. Advances in Computer Vision and Pattern Recognition. Springer International Publishing, 2015.
- [19] Kaspar Riesen, Andreas Fischer, and Horst Bunke. Computing upper and lower bounds of graph edit distance in cubic time. *Int. Workshop on Artificial Neural Networks in Pattern Recognition*, 8774:129–140, 2014.
- [20] Louisa Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992.
- [21] Miguel A. Ferrer, Moises Diaz-Cabrera, and Aythami Morales. Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):667–680, mar 2015.
- [22] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro. MCYT baseline corpus: a bimodal biometric database. *IEEE Proceedings-Vision, Image and Signal Processing*, 150(6):395–401, 2003.
- [23] J. Fierrez-Aguilar, N Alonso-Hermira, G Moreno-Marquez, and Javier Ortega-Garcia. An off-line signature verification system based on fusion of local and global information. In *Biometric Authentication*, pages 295–306. Springer, 2004.
- [24] Miguel A. Ferrer, J. Francisco Vargas, Aythami Morales, and Aarón Ordonez. Robustness of Offline Signature Verification Based on Gray Level Features. *IEEE Transactions on Information Forensics and Security*, 7(3):966–977, jun 2012.
- [25] Julian Fierrez-Aguilar, Javier Ortega-Garcia, and Joaquin Gonzalez-Rodriguez. Target dependent score normalization techniques and their application to signature verification. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(3):418–425, 2004.
- [26] F. Alonso-Fernandez, M.C. Fairhurst, J. Fierrez, and J. Ortega-Garcia. Automatic measures for predicting performance in off-line signature. In *Proc. 14th Int. Conf. on Image Processing*, pages 369–372, 2007.
- [27] Almudena Gilperez, Fernando Alonso-Fernandez, Susana Pecharroman, Julian Fierrez, and Javier Ortega-Garcia. Off-line signature verification using contour features. In *Proc. 11th Int. Conf. on Front. in Handwriting Rec.*, pages 1–6, 2008.