Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 26 (2005) 2537–2548

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Recognition-directed recovering of temporal information from handwriting images

Christian Viard-Gaudin *, Pierre-Michel Lallican, Stefan Knerr

University of Nantes, IRCCyN UMR 6597, Rue Christian PAUC, BP 50609, 44306 NANTES Cedex 3, France

Received 27 May 2004; received in revised form 4 March 2005
Available online 8 August 2005

Communicated by C.Y. Suen

Abstract

This paper analyses a handwriting recognition system for offline cursive words based on HMMs. It compares two approaches for transforming offline handwriting available as two-dimensional images into one-dimensional input signals that can be processed by HMMs. In the first approach, a left–right scan of the word is performed resulting in a sequence of feature vectors. In the second approach, a more subtle process attempts to recover the temporal order of the strokes that form words as they were written. This is accomplished by a graph model that generates a set of paths, each path being a possible temporal order of the handwriting. The recognition process then selects the most likely temporal stroke order based on knowledge that has been acquired from a large set of handwriting samples for which the temporal information was available. We show experimentally that such an offline recognition system using the recovered temporal order can achieve recognition performances that are much better than those obtained with the simple left–right order, and that come close to those of an online recognition system. We have been able to assess the ordering quality of handwriting when comparing true ordering and recovered one, and we also analyze the situations where offline and online information differ and what the consequences are on the recognition performances. For these evaluations, we have used about 30,000 words from the IRONOFF database that features both the online signal and offline signal for each word.

Keywords: Offline handwriting; Online handwriting; Graph modeling; Word recognition; Hidden Markov Model

1. Introduction

The handwriting recognition systems presented in the literature can be classified into offline

* Corresponding author. Tel.: +33 2 40 68 30 40; fax: +33 2 40 68 32 32.
E-mail address: christian.viard-gaudin@polytech.univ-nantes.fr (C. Viard-Gaudin).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.04.019

systems, where data is available as an image of the handwritten document, and online systems where temporal information collected during the writing process is available. Many factors differentiate these two domains: application areas, digitization process, data formats used and richness of information regarding the handwriting. The most prominent applications for offline recognition systems are bank check processing (Knerr et al., 1998), form processing (Cracknell and Downton, 1998) and postal address reading (Srihari and Keubert, 1997). In these situations, an already existing document is converted into an image using a camera or a scanner. The data is then available as a two-dimensional bitmap that contains some handwriting together with whatever was printed on the document before or after the handwriting: lines, boxes, combs, stamps, pictures, etc., and together with whatever noise is generated by the digitizing process. A first difficulty is to be able to correctly locate and extract the handwriting from the background and other areas of interest such as graphics and photos. Once the ink pixels are available, it is possible to know the position of the strokes, but not the temporal order/direction in which they were written. All the dynamic information related to how handwriting has been produced is lost with offline data. This is why, it is also referred to as a static recognition problem. Many office tasks will likely take advantage of the possibility of converting paper-based information into the digital world. With online handwriting systems, the data is captured while the writing is in progress, and becomes available as a sequence of points sampled along the trajectory of the writing tool. In such a situation, dynamic information is derived from the sequence of coordinates (x_n, y_n), and gives additional clues on the ductus[1] of the writing. Pen-Up and Pen-Down positions, and stroke orientation are readily available from this point sequence. This information allows online features to be used in order to segment overlapping words and characters more easily than with only the static image data. Conversely, online features may also introduce ambiguities and unwanted variations, specifically with delayed strokes and other information irrelevant for recognition process (e.g., whether a '1' is written with a stroke oriented downward or upward). Of course, offline features are not sensitive to this problem. Therefore, online and offline features complement each another, and much research work has been dedicated to combine both features for improved recognition. For complementing online recognition systems with offline features, two main strategies are commonly employed. Either the feature set extracted from the strokes is enriched with offline features (Tanaka et al., 1999) and a single recognizer processes the extended vectors, or the combination (Kittler et al., 1998; Suen and Lam, 2000) is performed at the classifier level with two different classifiers. The online input data is first recognized by a pure online system, and also converted into off-line bitmap and recognized separately by an offline classifier (Vinciarelli and Michael Perrone, 2003). The complementarity of the offline data with respect to the online signal was shown to improve recognition accuracy. Generating offline data from the available online information is fairly straightforward, even though some parameters have to be set empirically: width of the strokes, width variations, and interpolation algorithms.

The problem of complementing an offline recognition system with online features is quite different and much more challenging. Recovering dynamic information from static images is not straightforward. The work presented in this paper is an attempt at recovering the temporal order and orientation of strokes in order to improve the recognition rate of an offline recognition system. Our goal is not to reconstruct the true pen trajectory but to increase the stability of the description of handwriting for a better recognition. In that sense, the proposed approach, termed *REC–REC*, combines the temporal RECovering stage with the RECognition stage. Although some other attempts have been made to recover temporal information from static images, we are not aware of any other research work where temporal recovering and recognition are cooperating. Most of the time, the evaluation is limited to a visual ad hoc appreciation on a limited number of specific configurations (Doermann and Rosenfeld, 1995),

[1] Ductus: this Latin word defines the stroke order and stroke orientation in calligraphy.

sometimes even restricted to single-stroke handwriting (Kato and Yasuhara, 2000). However, some authors proposed algorithms that were tested for classifying handwritten words (Bunke et al., 1997; Jäger, 1998). Unfortunately, no feedback on the quality of the recovering stage has been given. In the latter paper, online recognition rate based on the recovered online information was about 73%, which is far below the recognition rate computed by a purely offline recognition system (94%) developed at Daimler-Benz (Caesar et al., 1994), both trained and tested on the same postal address databases. In our work, we achieve a 87% recognition rate with the baseline offline recognition system, termed *SCAN–REC* and presented in a subsequent section of this paper, and we were able to increase this rate to 92% by using the *REC–REC* recognition system. That means that the pseudo-online recognizer outperforms the bare offline recognizer. A comparison with results obtained with a real online system, termed as *ON–REC*, is also provided. These results are obtained on the IRONOFF database, briefly introduced in Section 4.1.

Section 2 is dedicated to the presentation of the global recognition system. The graph modeling approach that is used to recover the ordering and the orientation of the strokes is presented in Section 3. A detailed discussion of the various experiments that we have conducted is given in Section 4. In addition to the comparison of the recognition results obtained with the three proposed systems (*SCAN–REC*, *REC–REC*, *ON–REC*), we also analyze the situations where offline and online information differ and what the consequences are on the recognition performances.

2. The recognition system

The core of the recognition system is based on discrete Hidden Markov Models (HMMs). While more sophisticated architectures have been developed (Plamondon and Srihari, 2000) and proved their interest in terms of recognition rate, the point here is to focus on the complementarity between online and offline features for designing an offline recognition system. Subsequently, we have used a relatively standard recognizer, and we investigate its behavior with respect to three observation sequences resulting from (i) a left–right scan of the word—referred as *SCAN–REC* further, (ii) a time order of the strokes recovered previously from the static image—referred latter as *REC–REC*, (iii) a time order of the strokes corresponding to the true online ordering—referred as *ON–REC*.

The question that we want to address with this work, is what is relevant in the temporal information that is embedded within handwriting. With *SCAN–REC*, we remove all this temporal information; with *REC–REC*, we keep—at least we try to keep—the temporal information that is visible from the ink traces; whereas with *ON–REC*, we use exactly the real ordering of strokes. This is the real novelty of the paper, generally little attention is given to the segmentation process which allows to build the observation sequence explained by the HMMs. By focusing on how the observation sequence is built, we would like to bridge the gap between online and offline handwriting and provide some elements concerning the strengths and weaknesses of these two domains.

The overall structure of the recognition system is illustrated in Fig. 1.

2.1. SCAN–REC system

Within the first approach, termed *SCAN–REC* system, we use a standard sliding window technique to derive the frame sequence from the word image, see Fig. 2. The width of the window is normalized to one third of the size of the core zone

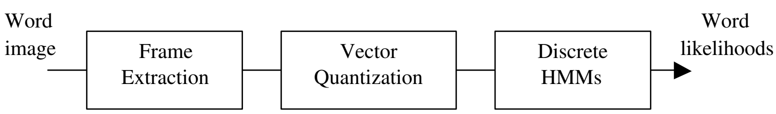

Fig. 1. Overall structure of the recognition system.

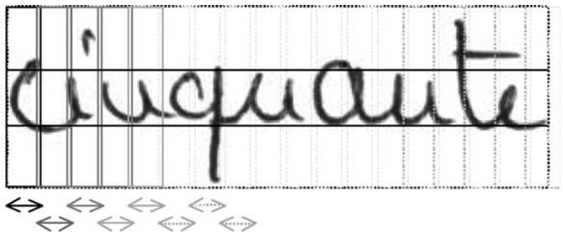

Fig. 2. Left-to-right sliding window segmentation.

that is first automatically extracted (Bengio and LeCun, 1994). Then, on each slice, a set of 47 features is computed. Among them, 7 are global features whereas the 40 others are computed locally on 10 different parts of the windows. The resulting description of the handwritten word is a sequence of frames, each frame being composed of a vector with 47 components.

2.2. REC–REC system

In the second approach, termed *REC–REC* system, the aim is to recover the time order of a stroke-based description, computed from the offline image. Considering the example given in Fig. 3, we would like to obtain the following sequence: Pen-Down–S7–S3–S2–S6–S5–S1–S1–S5–S8–Pen-Up–Pen-Down–S4–Pen-Up.

In this description, individual stroke segments S*i* has been previously obtained using an approach that segments the handwriting image into regular and singular parts (Lallican and Viard-Gaudin, 1999). Each stroke segment is oriented in order

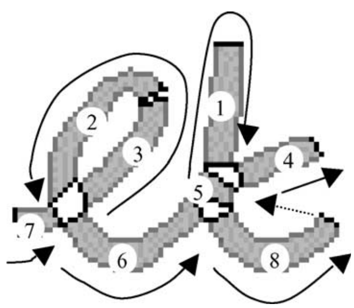

Fig. 3. Reconstructed ductus from the offline image: Pen-Down–S7–S3–S2–S6–S5–S1–S1–S5–S8–Pen-Up–Pen-Down–S4–Pen-Up.

to model the true writing process. In the previous sequence, the subsequence S5–S1–S1–S5 explicitly models a stroke segment that has been drawn once upward and next downward.

As for the window-based segmentation, a feature vector is extracted from each stroke segment S*i*. Each stroke segment being a small component, a small number of seven geometrical features is enough to characterize every type of stroke. At this point, the offline-handwritten word is described by a sequence of feature vectors. The remaining part of the recognition process is the same for both methods used to compute the observation sequence.

The next step consists in representing each frame by a unique symbol in order to produce an observation symbol sequence. We perform a vector quantization of the feature space using the K-means algorithm and a simple Euclidean metric. We have used typically $K = 300$ clusters. Fig. 4 shows examples of five different clusters.

The recognition of the resulting symbol sequences is achieved by using discrete HMMs. We have used 54 left–right letter models (a–z, A–Z, –, "dia") with a number of states proportional to the average number of observations constituting the corresponding letter, "dia" designating a generic model for diacritical marks. The topology model

Fig. 4. Each row shows some examples of segments belonging to the same cluster. For each row, the two horizontal lines indicate the core zone, which is automatically extracted from the word image. The temporal orientation of the stroke is given by the two colors of each segment: light gray first.

(self loops and state transitions) is the same for all letter models and has been defined empirically.

A word model is a left–right concatenation of letter models with optional diacritic models inserted between letters. This topology of the word models is usable for any Latin writing, but it is particularly suited to the French language, which shows a large number of diacritical marks that can be written any time after the letter to which they belong. Diacritical marks include "i" and "j" dots, accents such as in letters as "é", "à", "û", and "t" bars.

Fig. 5 presents the topology of the French word "cintre". This word contains two diacritical marks: an "i" dot and a "t" bar. The "i" dot can be written just after the "i" or after the "n", which is modeled by the first two dia-models without self transition. From letter "t" on, it is possible that the two diacritical marks are written consecutively. This is modeled by the loop over the diacritical mark model "dia".

2.3. ON–REC system

The *ON–REC* system is nearly the same as the previous one. The same steps of segment extraction from the offline image, feature extraction from the segments, and then *K*-means clustering to assign a symbol to segments are carried out. The only difference is the ordering of the segments. With *ON–REC*, we use the true ordering of the segments as they are provided by the online part of the IRONOFF database.

For the three systems described above, the HMMs have been trained using the Baum–Welch training algorithm. The word likelihoods can be computed by the forward–backward algorithm where the lexicon is either flat or organized in a trie structure (Rabiner, 1989).

3. A graph modeling approach

The "*REC–REC*" approach proceeds by a cooperative process between the recovering stage, which intends to recover the temporal order of the segments extracted from the word image, and the recognizing stage itself. A graph modeling approach described in details in (Lallican et al., 2000), is used to recover the temporal stroke ordering. It results in a list of recovered ordering candidates that is established by exploring two graph representations: a graph at the segment level and a graph at the stroke level.

Two searches are performed in these two different graphs: the first search finds a set of strokes (handwriting between Pen-Down and Pen-Up), while the second search orients and orders the strokes. Figs. 6 and 7 present the two graphs corresponding to the word "et" displayed in Fig. 3.

3.1. Recovering the intra-stroke order

We can note on the graph presented in Fig. 6a that (i) every segment Si is represented by two nodes i' and i'' connected with an intra-link (dark gray edge), (ii) inter-links (light gray edge) connect nodes which share a common connected area in the image, and (iii) a last type of link, extra-links displayed in dotted lines, corresponds to edges of completion. The completion of the graph is required in order to transform the search for a pre-Hamiltonian path—each node being visited at least once—in the search for a Hamiltonian path where every node is visited exactly once. Concerning the cost function, we have used appropriate local continuity criteria including curvature and width stroke preservation.

There are algorithms that can compute good approximations of the optimal path, we used one

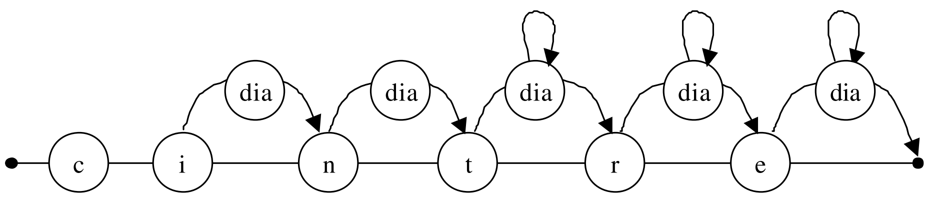

Fig. 5. Model for the French word "cintre".

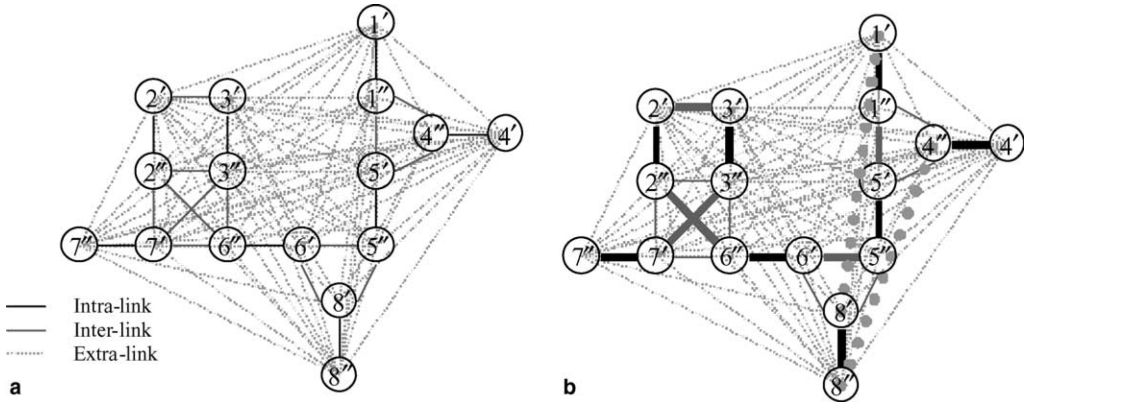

Fig. 6. First graph at the segment level: (a) the completed graph of the word "et" and (b) optimal path for the first optimization process.

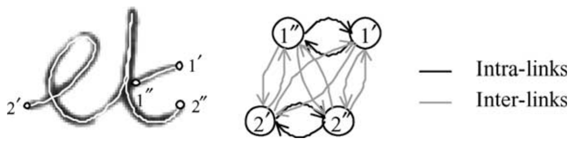

Fig. 7. Second graph model for recovering stroke orientations and stroke ordering.

based on a Tabou method (Gondran and Minoux, 1984), which appears quite efficient. The optimal path proposed by our search algorithm is displayed in Fig. 6b, it consists of the sequence: 7′–7′–3″–3′–2′–2″–6″–6′–5″–5′–1″–1′–8′–8″–4″–4′. Here, 1′–8′ is a retracing and 8″–4″ is a Pen-Up/Pen-Down movement.

3.2. Recovering the inter-stroke order

The second optimization process aims at recovering the orientation of each stroke and the time order of the strokes found in the first optimization process. Therefore, the entities manipulated in the second graph model are strokes that play the same role as the segments in the first optimization process. A node is associated with each stroke extremity, and intra-links characterize the stroke orientation while inter-links define the stroke order. Note, that in contrast to the first graph model, the second optimization process uses directed graph models, see Fig. 7.

The number of possible paths of the "*REC*–*REC*" approach for a word with N segments is $2N!$ We limit the number of ordered candidates to the $N1$ best candidates in the first graph, corresponding to $N1$ different segmentations into strokes. For each of these candidates, $N2$ stroke orders are proposed, corresponding to the $N2$ best paths of the second graph. Thus, $N = N1 \times N2$ ordered candidates are in competition in the recognition system. Fig. 8 shows an example of the four best recovered trajectories with $N1 = N2 = 2$ for the French word "sept".

Once the best path, or the N-best paths, have been extracted, the system computes likelihood for each word from the lexicon along these N-best paths using the HMMs presented in the previous section. The combination path/word with the highest likelihood will be selected as the recovered sequence and the recognized word.

4. Experiments and performance comparisons

4.1. Dual handwriting database: IRONOFF

For the experiments reported in this paper we have used a training set of 20,898 words and a test set of 10,448 words from a 197 word lexicon (French and English). All data is taken from the IRONOFF dual database that has been presented in more detail in (Viard-Gaudin et al., 1999) and which was collected from among approximately 700 writers. For each word in the database,

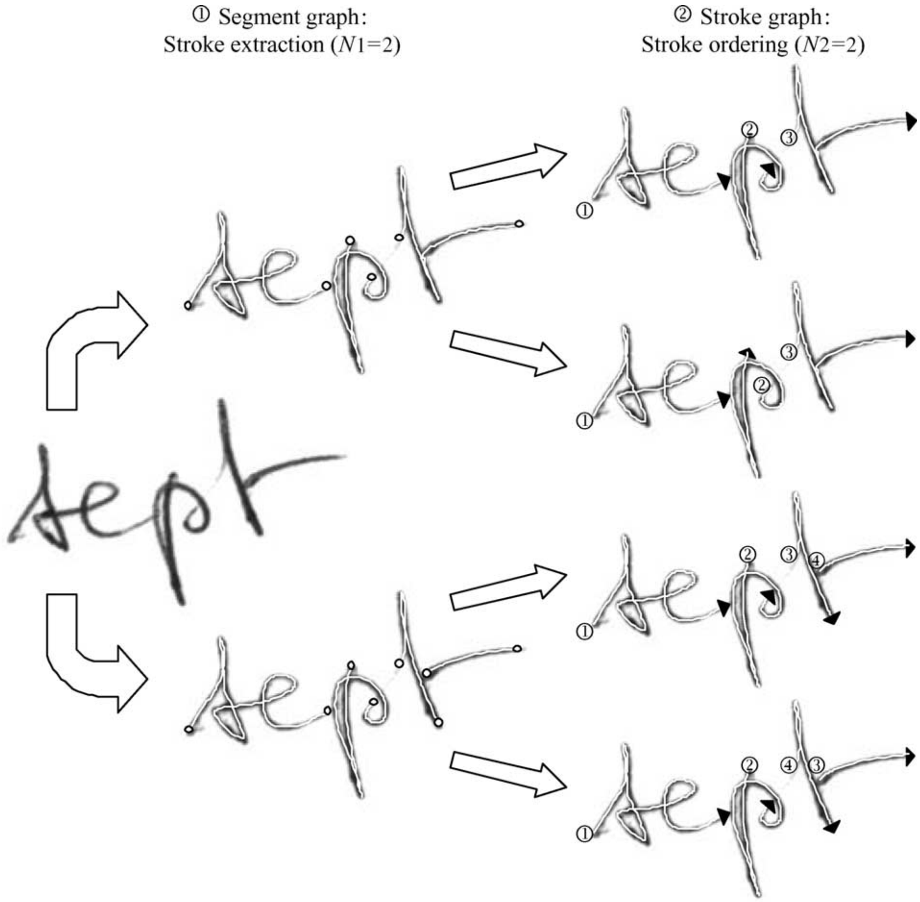

Fig. 8. Stroke ordering hypothesizes for the French word "sept" (seven). (○) Pen-Down and Pen-Up; (①②③④) Pen-Down and stroke ordering; (▶) Pen-Up.

Fig. 9. Examples from the IRONOFF database showing the offline and the online signal (left: linear interpolation between online points; right: online points only).

IRONOFF provides the offline pixel image scanned with a resolution of 300 dpi, as well as the online signal which was sampled at 100 points per second on a Wacom UltraPad A4. Fig. 9 shows two examples with the offline image in the background and the corresponding online points in the foreground. With IRONOFF database, offline data are not just a synthesized image computed from the online points but result from the scanning of a word written with an ink pen.

4.2. Recognition rate comparison

Fig. 10 shows the recognition rate on IRONOFF test set with respect to the number K of candidates in the top list and with different strategies for obtaining the observation sequence processed by the HMMs. With the baseline *SCAN–REC* system, a recognition rate of 87.4% is achieved in top one position. Introducing the *REC–REC* approach allows to reach 89.8% when we consider only one time ordered ($N = 1$) sequence, which corresponds to the best path extracted from the two ordering graphs. If we extend the number of paths to the six best paths in each of the two ordering graphs, then recognition rate increases to 92.0%. It is a significant 37% reduction in the error rate with respect to the *SCAN–REC* system. No more improvement was achieved by increasing the number of allowable paths beyond six for defining the time order sequence. By adding new paths, we give chances to recover more reliable paths than the top one extracted from the graphs. When such a path is found, the likelihood score will be better, and this yields to an increase of the recognition rate. However, at the same time, all other word models, including the true one, are competing. Consequently, every new path can increase not only the true word likelihood, but also other models, resulting in a possible recognition error.

As with the dual IRONOFF database the true order of handwriting strokes is available, we can use the true ordering sequence, instead of the sequence

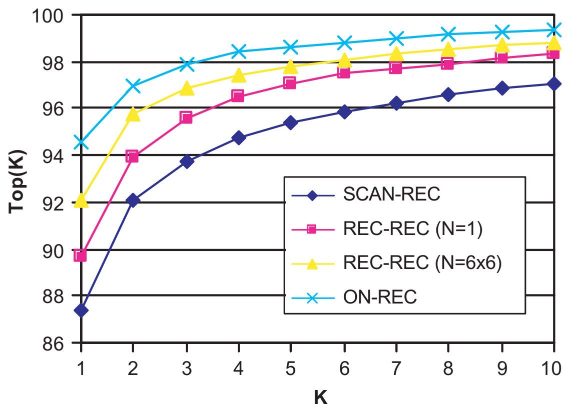

Fig. 10. Recognition rate as a function of the number of candidates for different strategies of constructing the observation sequence.

proposed by the *REC–REC* system. It corresponds to the so-called *ON–REC* system, where the graph approach is no more used: the sequence of segments is directly derived from the time information. It exhibits a 94.5% recognition rate. It is obvious that the reconstructed sequence does not perform as well as the genuine one, even when several paths are competing together. This is not a surprise since some information is definitively lost in the static image, and many experiments have shown superiority of online recognition compared to offline recognition (Seiler et al., 1996). Nevertheless, the results presented in Fig. 10 show that the offline system we introduced was considerably improved by the *REC–REC* approach and we are close to what seems to be the upper limit of this approach given by the true online ordering. According to our experiments, a significant gap still exists in the recognition rates between the true ordering sequence (*ON–REC*) and the recovered one (*REC–REC*) (94.2% instead of 92%). It is fair to state that this gap could be reduced by improving the recovering algorithm (better cost functions, enhanced exploration of the graphs) but it is hard to consider that we would be able to outperform the reference ordering.

The recognition rate increases rapidly with the number K of candidates. This is an interesting point to consider for post-processing or coupling with language modeling approach (Perraud et al., 2003). For instance, the *REC–REC* system with $N = 6 \times 6$ limits the error rate to 3% as soon as we consider three candidates (size of lexicon being $L = 197$). From a computation point of view, we have used a flat implementation for computing the various likelihoods, that means that the recognition time is linear with respect to the size of the lexicon and the number of paths (N). The word recognition time is about 0.2 s, with $L = 197$ and $N = 1$ on a 2 GHz PC. It could be easily improved by factorizing the computation at the lexicon level and at the graph level.

4.3. Quality of the recovered temporal stroke order

In the next experiment, in order to evaluate the quality of the recovered stroke order, we have conducted a comparison between the word likelihoods for the true word model computed by both the *ON–REC* system and the *REC–REC* system. In these experiments, the same HMMs are used with the same training data, the only difference is the ordering of the segments extracted from the images of the test set. In the former case, *ON–REC*, the true ordering is used, while in the latter, *REC–REC*, the ordering of the segments and strokes result from the competition of the N best paths of the graphs. The chart presented in Fig. 11 shows the percentage of samples of the test set which have a recognition score computed with the *REC–REC* system which is respectively lower (REC < ON), equal (REC = ON) and higher (REC > ON) than the recognition score computed with the *ON–REC* system. For 78% of the samples, the true online ordering produces a higher likelihood than the recovered ordering when we take into account only the best path from the graphs ($N = 1$). For 10% of the test set, always with $N = 1$, exactly the same likelihood score is obtained with the two different ordering methods. That means, no doubt, that the same ordering has been proposed, i.e., the recovered ordering is exactly the same as the true online ordering. This value could appear as rather small, but do not forget that the number of segments in a word is typically a few tens, and that a single switch between two segments prevents the resulting sequence to be counted as being correctly recovered since the recognition score will not be exactly the same in such a situation. For the remaining 12% of samples, recovered ordering performs better than the original ordering. This situation occurs specifically when corrections have been inserted in the online handwriting, in this case the true ordering is not reliable and the reconstructed one appears more stable from the recognition point of view. This situation is illustrated with the example of Fig. 12.

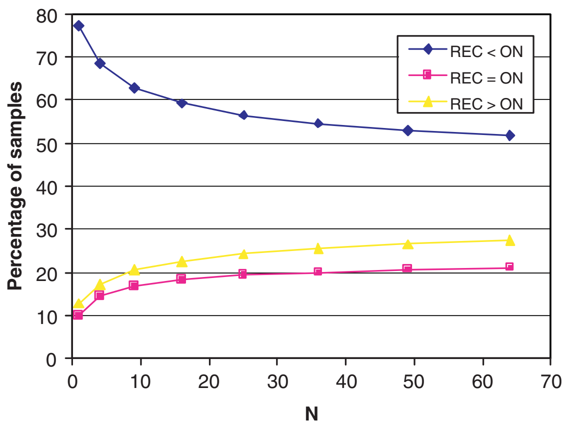

Fig. 11. Quality assessment of the recovered stroke order.

In the word "huit" (eight), the letter "i" has been written after the "t', and this delayed stroke alters the recognition score computed for the word "huit" whereas, with the proposed recovered order, the letter "i" has been inserted at its regular place, increasing significantly its log-likelihood score (−1.93 to −1.67). This is a desirable recovering error, since as already mentioned, the goal is not to recover necessarily the true ordering but a consistent ordering with respect to recognition. It is worth noting that as the number of competing paths increases, more and more samples are correctly reconstructed. We reach a limit of 21% of samples exactly reconstructed with $N = 8 \times 8 = 64$ paths, and 27% have a reconstructed sequence which is better than the true one. However, we cannot conclude that 27% of the words are affected by

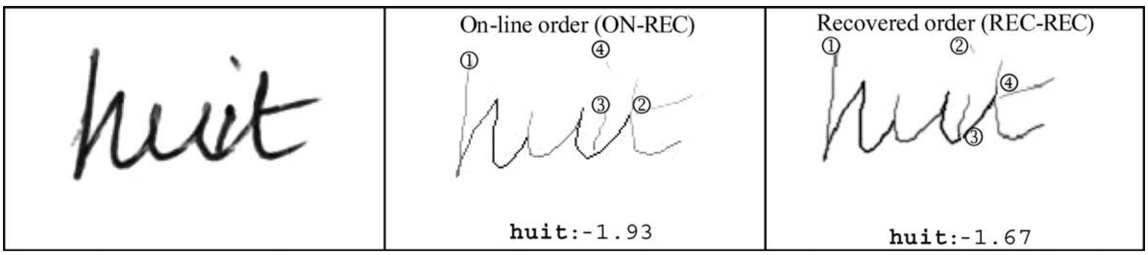

Fig. 12. Example where the recovered order is better than the true online order.

a writing correction—it would be hard to define objectively when this happens. In addition, there is a second cause that contributes to this percentage, it is due to non-standard writing styles as explained further in Section 4.4 There remains 52% of the samples that present a better true ordering than among any of the N paths resulting from the *REC–REC* approach. In most of these remaining cases, the score difference is very tiny, resulting in a single difference between the two orderings.

With this experiment where only the true word HMM is used, as can be seen in Fig. 11, adding new paths can only increase the number of words that take advantage of these possible sequences to outperform the true ordering. Of course, there is a limit, when all the "reasonable" paths have been proposed, there is no more hope to beat the true ordering. This limit appears to be around $N = 8 \times 8$, which is beyond the limit found in the recognition rate evaluation (Fig. 10, $N = 6 \times 6$) where all word-models were competing each others, and as mentioned previously, adding new paths may cause possible error recognition.

4.4. Quality of the true temporal stroke order

How can we trust the true online order? As pointed out by the previous example, Fig. 12, sometimes true ordering can be corrupted, either by a correction, or more basically, with a non-conventional writing style, (e.g., whether a loop is written clockwise or counterclockwise), which can increase the complexity of the modeling task and can lead to recognition errors. Precisely, we have examined the behavior of samples, which are not correctly recognized by the true ordering (*ON–REC*), when they are submitted to the *REC–REC* system. As depicted by Fig. 13, more than one third (37%) of these misclassified words with the *ON–REC* system are correctly classified with the *REC–REC* system when we consider $N = 8 \times 8 = 64$ paths. This is another manifestation of the complementarity between online and offline representations. In this case, online fails, but the recovered ordering from offline succeeds for more than one third of these cases. Such an example is displayed in Fig. 14. The true ordering is clockwise for the first letter "Q", which is not a common style, but leads to rank correct word "Quiz" in second position, whereas the *REC–REC* system assumes a counterclockwise orientation for the first stroke, this being more favorable according to what models have been learnt during training, and consequently, the log-likelihood score increases from −2.51 to −2.36, giving top position for the true label.

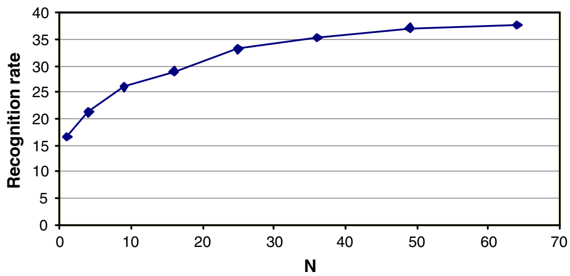

Fig. 13. Recognition rate with *REC–REC* system on misclassified words with *ON–REC*.

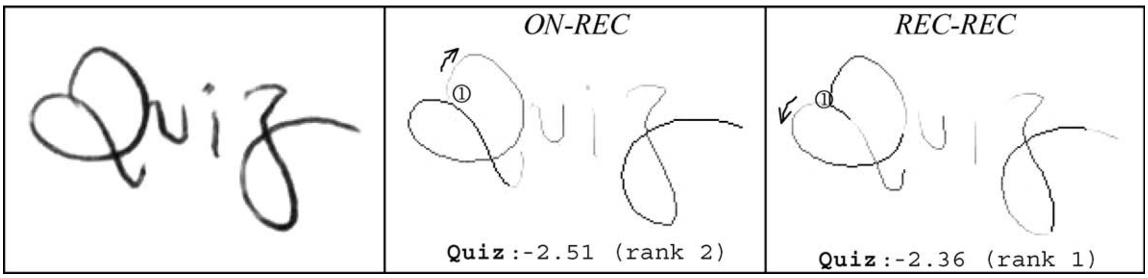

Fig. 14. Example illustrating an unusual online writing style.

5. Conclusion

We have demonstrated that it is possible to model and often recover the temporal stroke ordering in static handwritten words using a graph-based approach. Our approach first generates a ranked list of possible stroke orderings by segmenting the word image into candidate strokes and possible temporal sequences thereof. These sequences aim at mimicking the online stroke ordering of natural handwriting in such a way that our offline recognition system can take advantage of online properties for improving the recognition accuracy. Then, our recognizer selects the most likely temporal stroke ordering based on a training process on a large set of handwriting samples for which the temporal information was available. Using the dual IRONOFF database, we have shown that online recognition systems can achieve higher recognition rates than offline recognition systems. But our experiments have also reinforced the idea that offline representation and online representation of handwriting can be complementary, and that offline data representations can help online recognition systems to achieve higher accuracy because they provide a description that is independent of unusual stroke orders. A word recognition rate of 87.4% has been obtained on a test set without any attempt to recover the time-order of writing (*SCAN–REC* system). Using only the top candidate for the recovered temporal stroke order, the recognition rate increases to 89.8% (*REC–REC*, $N = 1$), and finally using recognition to control and select the best recovered stroke order, we achieve a 92.4% recognition rate (*REC–REC*, $N = 6 \times 6$). This result comes close to the recognition rate of 94.4% that is obtained when using the true online ordering (*ON–REC* system). Our analysis of the quality of the proposed recovering scheme indicates that nearly 50% of the proposed recovered stroke orders are at least as good as the true online ordering with respect to the recognition goal. We have found that the true online stroke order is deficient in more than one third (37%) of the words that have been misclassified by our online recognition system. Hence the interest in combining offline features and online features for online recognition systems. Besides handwritten word recognition, recovering the stroke order can also be useful for other tasks such as signature verification. Lau et al. (2003) have initiated promising work on this subject.

References

Bengio, Y., LeCun, Y., 1994. Word normalization for on-line handwritten word recognition. In: Proc. Internat. Conf. on Pattern Recognition. pp. 409–413.

Bunke, H., Ammann, R., Kaufmann, G., Ha, T.M., Schenkel, M., Seiler, R., Eggimann, F., 1997. Recovery of temporal information of cursively handwritten words for on-line recognition. In: Proc. 4th Internat. Conf. on Document Analysis and Recognition (ICDAR 1997). pp. 931–935.

Caesar, Y., Gloger, J.M., Kaltenmeier, A., Mandler, E., 1994. Recognition of handwritten word images by statistical methods. In: 1st IEE Workshop Handwriting Analysis and Recognition: A European Perspective, Brussels.

Cracknell, C., Downton, A.C., 1998. A handwritten form reader architecture. In: Proc. 6th Internat. Workshop on Frontiers in Handwriting Recognition (IWFHR'98). pp. 67–76.

Doermann, D.S., Rosenfeld, A., 1995. Recovery of temporal information from static images of handwriting. Internat. J. Comput. Vision 15, 143–164.

Gondran, M., Minoux, M., 1984. Graphs and Algorithms. Wiley, New York.

Jäger, S., 1998. Recovering dynamic information from static, handwritten word images. Ph.D. thesis, University of Freiburg.

Kato, Y., Yasuhara, M., 2000. Recovery of drawing order from single-stroke handwriting images. IEEE Trans. Pattern Anal. Machine Intell. 28 (9), 938–949.

Kittler, J., Hatef, M., Duin, R., Matas, J., 1998. On combining classifiers. IEEE Trans. Pattern Anal. Machine Intell. 20 (3), 226–239.

Knerr, S., Augustin, E., Baret, O., Price, D., 1998. Hidden Markov model based word recognition and its application to legal amount reading on French checks. Special issue on Document Image Understanding of the Journal for Computer Vision and Image Understanding.

Lallican, P.M., Viard-Gaudin, C., 1999. Off-line handwriting modeling as a trajectory problem. In: Advances in Handwriting RecognitionSeries in Machine Perception and Artificial Intelligence, vol. 34. Word Scientific, Singapore, pp. 449–458.

Lallican, P.M., Viard-Gaudin, C., Knerr, S., 2000. From offline to on-line handwriting recognition. In: Proc. 7th Internat. Workshop on Frontiers in Handwriting Recognition. pp. 303–312.

Lau, K.K., Yuen, P.C., Tang, Y.Y., 2003. Recovery of writing sequence of static images of handwriting using UWM. In: Proc. 7th Internat. Conf. on Document Analysis and Recognition (ICDAR 2003). pp. 1123–1127.

Perraud, F., Viard-Gaudin, C., Morin, E., Lallican, P.M., 2003. *N*-gram and *N*-class models for on-line handwriting recognition. In: Proc. 7th Internat. Conf. on Document Analysis and Recognition (ICDAR 2003). pp. 1053–1157.

Plamondon, R., Srihari, S.N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intell. 22 (1), 63–84.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Read. Speech Recogn., 267–296.

Seiler, R., Schenkel, M., Eggimann, F., 1996. Off-line cursive handwriting recognition compared with on-line recognition. In: Proc. 13th Internat. Conf. of Pattern Recognition (ICPR 1996), vol. 4. pp. 505–509.

Srihari, S.N., Keubert, E.J., 1997. Integration of handwritten address interpretation technology into the United States postal service remote computer reader system. In: Proc. 4th Internat. Conf. on Document Analysis and Recognition (ICDAR 1997). pp. 892–896.

Suen, C.Y., Lam, L., 2000. Multiple classifier combination methodologies for different output levels. In: Proc. Internat. Workshop on Multiple Classifier Systems. pp. 52–66.

Tanaka, H., Nakajima, K., Ishigaki, K., Akiyama, K., Nakagawa, M., 1999. Hybrid pen-input character recognition system based on integration of online–offline recognition. In: Proc. 5th Internat. Conf. on Document Analysis and Recognition (ICDAR 1999). pp. 209–212.

Viard-Gaudin, C., Lallican, P.M., Knerr, S., Binter, P., 1999. The IRESTE on–off (IRONOFF) handwritten image database. In: Proc. 5th Internat. Conf. on Document Analysis and Recognition (ICDAR 1999). pp. 455–458.

Vinciarelli, A., Michael Perrone, M., 2003. Combining online and offline handwriting recognition. In: Proc. 7th Internat. Conf. on Document Analysis and Recognition (ICDAR 2003). pp. 844–848.