IJDAR (2014) 17:455–469
DOI 10.1007/s10032-014-0231-x

ORIGINAL PAPER

Handwriting synthesis: classifications and techniques

Yousef Elarian · Radwan Abdel-Aal · Irfan Ahmad · Mohammad Tanvir Parvez · Abdelmalek Zidouri

Received: 15 February 2014 / Revised: 12 July 2014 / Accepted: 15 September 2014 / Published online: 12 October 2014

Abstract Handwriting synthesis is the automatic generation of data that resemble natural handwriting. Although handwriting synthesis has recently gained increasing interest, the area still lacks a stand-alone review. This paper provides classifications for the different aspects of handwriting synthesis. It presents the applications, techniques, and evaluation methods for handwriting synthesis based on the several aspects that we identify. Then, it discusses various synthesis techniques. To the best of our knowledge, this paper is the only stand-alone survey on this topic, and we believe it can serve as a useful reference for the researchers in the field of handwriting synthesis.

1 Introduction

Handwriting synthesis, also referred to as *synthesis* hereafter, refers to the artificial generation of data that resemble human writing. Synthesis has applications such as the improvement of text recognition systems, PC-personalization, forgery detection, and Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). These applications may require certain specifications on the synthesized data or on the synthesis technique, such as being of a specific writer's style or a specific script. These applications also suggest methods to evaluate the adequacy of synthesized data.

Handwriting can be modeled via the simulation of the human writing process (*top-down* approach) or of its outcome (*bottom-up* approach). In the top-down approach, the neuromuscular acts of writing are simulated in what is commonly termed *movement simulation*. When the data itself is regenerated without imitating human movements, synthesis is termed *shape simulation* [1]. Moreover, there are online and offline handwriting synthesis scenarios. In online synthesis, text is generated with temporal information. On the other hand, text is generated without temporal information but with stroke thickness and inking information in offline synthesis.

Synthesis can be seen as the reverse of more well-known problems. For example, when synthesis aims at the generation of individual characters from their ASCII codes, it can be regarded as the reverse of character recognition. Similarly, when synthesis aims at the generation of words through the concatenation of characters, it can be regarded as the inverse of the character segmentation problem.

Handwriting synthesis has recently become a hot topic with increasing interest from the research community. Among the refereed journals that contribute to the dissemination of established knowledge in the area are the International Journal of Document Analysis and Recognition (IJDAR) (e.g., [2–4]), Pattern Recognition (e.g., [5–8]), Pattern Recognition Letters (e.g., [9]), Machine Learning (e.g., [10]), and others. Besides, some prestigious conferences such as the International Conference on Document Analysis and Recognition (ICDAR) (e.g., [11–14]), the International Workshop

Y. Elarian (✉) · R. Abdel-Aal · I. Ahmad · A. Zidouri
King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia
e-mail: yarian@kfupm.edu.sa

R. Abdel-Aal
e-mail: radwan@kfupm.edu.sa

I. Ahmad
e-mail: irfanics@kfupm.edu.sa

A. Zidouri
e-mail: malek@kfupm.edu.sa

M. T. Parvez
Qassim University, Qassim 51452, Saudi Arabia
e-mail: m.parvez@qu.edu.sa

on Document Analysis Systems (DAS) (e.g., [15]), the International Conference on Pattern Recognition (ICPR) (e.g., [16–19]), and the International Conference on Frontiers in Handwriting Recognition (ICFHR) (e.g., [20–23]) help in spreading the advances in the field. However, there is a lack of a stand-alone paper that surveys the field. Hence, we contribute in this paper by classifying the applications of synthesis and linking them to certain aspects of synthesis. We also survey the synthesis techniques with more focus on shape-simulation approaches.

The rest of this paper is organized as follows: In Sect. 2, we link the applications, aspects, and evaluation methods of synthesis. Then, we present a review of shape-simulation approaches in Sect. 3. In Sect. 4, we present other techniques. Finally, we summarize and conclude in Sect. 5.

2 Synthesis applications, aspects, and evaluation methods

The applications of synthesized data guide the aspects of synthesis and dictate methods to evaluate their results. In this section, we identify some handwriting synthesis applications and link them to the aspects and evaluation methods that may suit them.

2.1 Synthesis applications

Handwriting synthesis has a wide range of applications. It can be used to generate desired and inexpensive ground-truth data for the development of text segmentation and recognition systems [24]. Writer-specific synthesis can also be a means for font personalization [25,26], calligraphy generation, word spotting, forgery detection, and writer identification. In addition, a recent application for synthesis is the production of CAPTCHAs.

Synthesized handwriting might target humans, machines, or both. It may be intended to imitate a particular writer's style, to generate writer-independent handwriting, or to tell humans and machines apart. Synthesized calligraphy, for example, targets human subjects [21,27,28], whereas generic training data may target text recognition systems [9,15,29]. Word spotting systems may benefit from writer-specific synthesis to find words written by a particular scribe [30,31] and from generic synthesis to find words regardless of scribes. CAPTCHA requires reasonable human legibility but low machine readability [32].

Figure 1 shows some applications on a machine/human readability plane. Handwritten CAPTCHAs exploit the gap between humans and machines in reading handwriting [5]. Calligraphic and personalized fonts aim at the aesthetic aspects of writing but may be confusing to machines. On the other hand, some perturbed and noisy text which might not be pleasant to humans can be useful for training recognition systems [13,17,29]. Steganography, the art of hiding data, is a possible application for synthesized handwriting where secret messages can be communicated by certain choices of the optional features in a script [33,34].

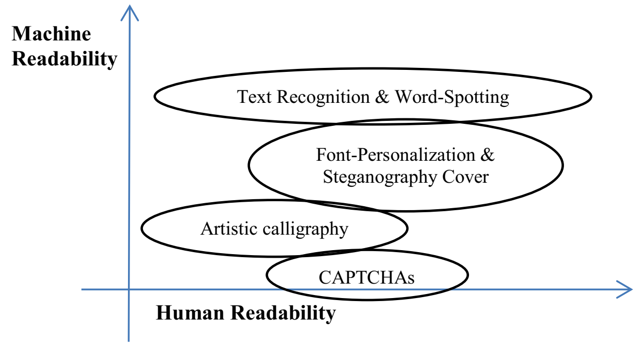

Fig. 1 Applications of handwriting synthesis on the human- versus machine-readability graph

2.2 Aspects of handwriting synthesis

The following aspects of handwriting synthesis can be specified based on the needed applications into:

1. *Level of granularity:* Strokes, characters, character groups/sub-words, words, lines or paragraphs
2. *Techniques:* Generation *or* concatenation.
3. *Online versus offline data types*
4. *Scripts and languages:* Arabic, Chinese, Indian, Latin, etc…
5. *Writing variability:* Writer-specific vs. writer-independent
6. *Parameterization:* parametric versus non-parametric system (this is for a system not data)

Most of the aspects above specify the outputs of a synthesis system. The techniques and the parameterization aspects specify synthesis systems, rather than their outputs. The level of granularity can qualify input or output data, but the aspect, here, is restricted to the synthesis output.

2.2.1 Level of granularity

The "level of granularity" refers to the unit (whether as strokes, characters, character groups, words, lines, or paragraphs) that is output by a synthesis system.

2.2.2 Techniques

Handwriting synthesis receives images of handwritten samples and generates new handwriting images. The input and output images can be of similar or different levels of granularity. Based on the relationship between the granularity levels of the input units and the output units, we classify synthesis techniques into two types: generation techniques and concatenation techniques. Generation techniques produce new synthesized images at the same level of the input samples they receive. Concatenation techniques, in contrast, produce output images at higher levels than their inputs, often guided by ASCII or Unicode text specifying the required synthesis string.

2.2.3 Online versus offline data

Online data, such as coordinate time-stamps and pressure, are captured as writing occurs on special devices called tablets. Figure 2a, c shows online data. Offline data are taken as static images of script that are usually written on paper. Offline data lack temporal information but contain inking and stroke-thickness information (e.g., Fig. 2d). The data types of the inputs and the outputs of synthesis systems are usually the same. Sometimes, however, online data might be used to generate offline-like outputs, often by the addition of inking effects [15,20,27]. In addition, some systems utilize a mixture of online and offline data in their inputs (e.g., Fig. 2b), for example, when a printed character is used as a standard reference for handwritten samples [35].

2.2.4 Scripts and languages

A script can be used to write several languages. The Latin script, for example, is used in English and Spanish languages. A script can be inherently cursive as in Arabic, inherently discrete as in Hiragana [36] and Katakana [37], or mixed as in modern Latin. Researchers have worked on synthesis of Latin [20,21], Arabic [4,38], Cyrillic [9], Chinese [11, 39], Korean (Hangul) [15], Japanese ([36,40] and [37]), and Indian (Hindi, Tamil, Malayalam, and Telugu) scripts [23]. Occasionally, systems are implemented and tested on multi-scripts [2,9,30].

2.2.5 Writing variability

Synthesis may or may not aim at the imitation of a specific writer's style, depending on the applications. Synthesis for character recognition improvement [15,36,41,42], as well as for CAPTCHA generation, usually lacks writer-specific features [5,43]. On the other side, applications such as PC personalization [20,21,28,44] and writer identification [28,45,46] call for writer-specific synthesis. In Table 1, we classify the applications of handwriting synthesis by their writer imitation and target aspects. In some cases (e.g., [16]), large databases of handwriting can be synthesized to generate writing samples for a single writer as well as in multi-writer

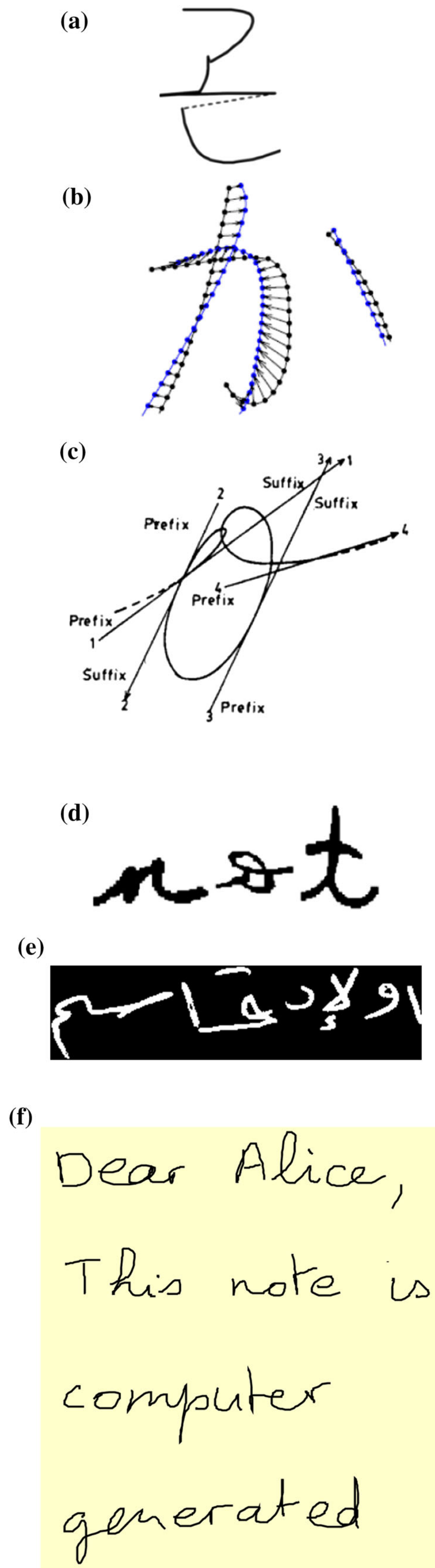

Fig. 2 Samples of synthesized handwriting at the **a**, **b**, **c** character, the **d** word, and the **f** paragraph granularity levels with different aspects

Table 1 Reported works on handwriting synthesis for the human and machine targets

Writing variability target	Writer independent	Writer specific
Human	Pen-based PC [1,21]	Writer imitation [38,48]
	Calligraphy arts [27,28]	PC personalization [20,22,27,28,44,49]
	CAPTCHAs [5,43]	
Machine	Text recognition [12,14,15,17,18,28,29,36,38,41,42,48,50–53]	Writer identification [14,28,46,48]
	Word spotting [38]	Word spotting [38]
	Compression [9,42]	

setup. Some researchers have developed systems that can function in either a writer-independent or a writer-specific modes [22,47].

2.2.6 Parameterization

Any synthesis technique uses a number of parameters which need to be learned or calibrated. Parameterization is an important aspect to study when comparing different synthesis techniques. Systems with fewer parameters are generally preferable. Parameters may also affect the computational efficiency of a technique. Another important aspect of parameters is their estimation/training. Some techniques may involve parameters which require expert knowledge for calibration while other parameters may be trained from the data available. Moreover, the number of parameters that need to be trained also places some constraint on the minimum data required to robustly train the model [54]. But sometimes, more parameters provide increased flexibility in deciding the desired quality and property of the synthesized text.

Parameters are in general closely tied to the underlying techniques. For example, Sigma-lognormal models [16,55], signal-based models [48], and spline-based models [21,42] depend on parameters for the definition of character shapes. Parameterization may be used to smooth joining ligatures between characters in concatenation systems [5,25]. In generative systems, changes to samples are controlled via parameters. For example, perturbation is added to samples, as in [14,29,56]. *Naturalness* can be parameterized, as in [44], where the relative distance from the printed sample and the nearness to handwritten sample is considered *naturalness*.

Figure 2 shows examples of different synthesis based on the aspects previously described. Figure 2a [47] and Fig. 2b [40] exemplifies the generation of Hangul and Hiragana characters, respectively, whereas Fig. 2c [38], Fig. 2d [20], Fig. 2e [38] and Fig. 2f [20] exemplifies the concatenation of a character, a word, a text-line, and a paragraph, respectively. Figure 2a, c, f uses and generates online data (where dotted lines indicate pen-rises); Fig. 2d, e shows samples of offline Latin and Arabic words synthesis; Fig. 2b matches online and printed (offline) samples to synthesize offline characters. Figure 2b shows an example of a parameterizable system where naturalness is parameterized via the distance from the printed font of the *ka* character (the origins of the small arrows) to the online handwritten version (pointed to by the arrows), and Fig. 2c shows four numbered *shape vectors* that define the splines to form a synthesized "o" character. Figure 2f is a clear example of writer-specific synthesis. Each shape vector has a direction indicated by its prefix (origin) and suffix (destination).

2.3 Evaluation methods

The choice of evaluation methods for synthesized data depends on the application domains for which the synthesis system is designed. Evaluations are used for following two broad purposes: to assess the quality (e.g., syntactic correctness, naturalness) of synthesis itself and to assess how well synthesis fulfills an application's end goal. For example, in the case of text synthesis for OCR improvement, there can be some evaluation methods for the assessment of the quality of synthesized text and how well it imitates the training data in a quality like its natural appearance. On the other hand, there can be evaluation methods for the assessment of the goal of the OCR application, say by measuring the increase in text recognition results due to the addition of synthesized training data.

Commonly used evaluation methods fall into two main categories: subjective and objective. S*ubjective* evaluation methods mainly rely on the opinion of human subjects. Sometimes, trained experts may be solicited to decide whether some handwriting belongs to a specific writer, whereas in other cases, evaluators only need to decide whether the writing looks natural or not.

Several researchers have used subjective methods for evaluating the synthesized handwriting. Subjective opinions of 21 English native speakers, that were not among the 15 writers of the database of [52], were used to evaluate the performance of their parameter calibration. For example, Guyon

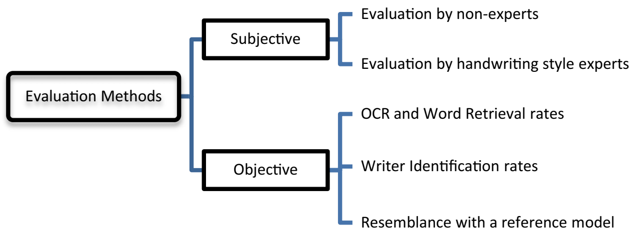

Fig. 3 Most common methods to evaluate synthesized data

mentioned that in subjective evaluation, the trained eye can find exaggerated regularities in letter shapes and probable inconsistencies in inking [20]. Other works that rely on subjective evaluation include [21,28,47].

Objective methods rely on quantitative measures for the evaluation of synthesized handwriting. Text and writer recognition systems give success rates which can be used as measures of the machine readability or writer resemblance of some handwriting [46,57]. In order to assess data that is synthesized for OCR improvements, the data can be injected to the training set. Injecting more synthesized data to training data is expected to improve the performance of the recognizer under the condition that the synthesized data captures the variability of natural writing. The premise is taken from a rule of thumb with real data: The more training data the better the recognition [58].

Figure 3 shows the most common evaluation methods grouped into the subjective and objective criteria.

Improvements in HMM-OCR performance on the IAM database were reported after the injection of synthetic training data in [13] and [57]. Support vector machine OCR that runs on a database of 10 Hiragana characters (from the HANDS-nakayosi t-98-09 database) was used in [15], with reported improvements on the OCR performance. Similar efforts for improving OCRs using synthesized data include [41,46,50,59]. A script recognizer was used to classify synthesized text into Arabic, Latin, or Russian by Vincent et al. [9]. Although all of their synthesized data were perfectly labeled with its correct script type, the authors commented that the differences between correlation coefficients were quite small and not very reliable. In [12], normal OCR Turing test is used for the evaluation of synthesized Arabic handwriting. The models derived in [48] achieve 99.4 % success rate when tested as recognizers.

Analysis by synthesis is an objective evaluation method that judges synthesizers by the quality of their recognition models. This evaluation method is especially useful with generative model-based synthesizers. An analysis by synthesis scenario was used in [48], where the authors performed a test of completeness on their statistical model to demonstrate the ability to recognize data not in the training set.

Another objective evaluation method for synthesis compares synthesized handwriting to some *reference model*. Dolinsky and Takagi consider printed Hiragana characters as reference models that are deformed by personal handwriting styles [60]. Correlations and regression analysis are used to quantify the difference between the synthesized and reference model. Zheng et al. [14] also quantify the amount of deformation needed for their fusion-based algorithm.

A combination of subjective and objective evaluations was performed by Rao [42]. He used his synthesis model to implement a recognition scheme, in an analysis by synthesis scenario. He also demonstrated the distances between some original and synthesized sample characters and reported the natural and legible appearance of the results. The results of character synthesis are reported to be similar to their corresponding natural characters. The shape vectors used in that work achieve 94 % success rate as recognition models.

The performance of CAPTCHAs is evaluated by low OCR recognition rates while preserving reasonable human legibility. Hence, both OCR and subjective evaluation methods are needed to evaluate CAPTCHAs [5,43].

2.4 Linking applications, aspects, and evaluation methods

Applications may drive specifications related to the aspects of synthesis systems. Table 2 suggests specifications of the outputs of synthesis systems for some common applications of synthesized handwriting along with some suitable evaluation methods. The script aspect is not shown because it directly follows from the application script.

Some applications require synthesized data that is constrained to the style of a specific writer, whereas other applications can accept (and sometimes require) writer-independent style. Examples of writer-specific applications include forgery detection and font personalization. Synthesized handwritten CAPTCHAs typically allow for mixed writing styles. For character recognition systems, it is important to synthesize texts having realistic variability of multiple writers. Artistic and calligraphic applications may be sometimes constrained to writer-specific styles and may sometimes benefit from several styles together.

Applications that target end users usually call for offline synthesis, for this is the natural representation for humans. On the other side, most applications that target computers, such as OCR and writer identification, may call for offline or online synthesis, based on their internal representation of data. The performance of these latter applications can be evaluated by objective measures that comply with their targets.

3 Review of shape-simulation approaches

Shape-simulation approaches for handwriting synthesis model the shapes of handwriting units rather than the move-

Table 2 Output specifications and evaluation methods for some common application

Application	Level of granularity	Online/offline data Type	Writing variability	Suitable evaluation methods
Word spotting	Word	Offline	Application dependent	Objective: retrieval accuracy/sensitivity rates
CAPTCHAs	Character string	Offline	Writer independent	Subjective/objective: human legible text with deteriorated OCR rate
Character recognition improvement	Text	Both	Usually not	Objective: recognition success ratio
				Objective: analysis by synthesis
Forgery detection	Words or text lines	Mostly offline	Writer-specific	Subjective: handwriting style experts
				Objective: writer identification results
				Objective: resemblance with a reference model
Calligraphic and aesthetic styles	Words or text lines	Offline	Style specific	Subjective: evaluation by experts and non-experts
Personalization	Words or text	Offline	Writer specific	Subjective: evaluation by non-experts
				Objective: writer identification results

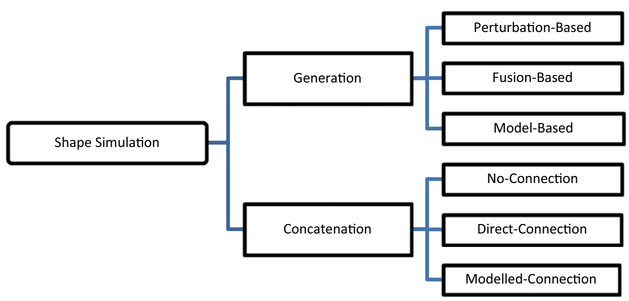

Fig. 4 Classification of shape-simulation synthesis techniques

ments that produce them. Shape-simulation techniques are practical when online data are not available, i.e., when data acquisition means are not restricted to PC tablets.

There are generation and concatenation techniques for shape simulation. Generation techniques synthesize new instances for a given writing unit, while concatenation techniques connect smaller scripting units into larger ones. Figure 4 shows a classification of shape-simulation techniques under the generation and the concatenation approaches.

Generation techniques can be subdivided into perturbation-based, fusion-based, and model-based techniques. Perturbation-based techniques alter one input sample to obtain new samples from it. Fusion-based techniques take two-to-few input samples and fuse parts of each into novel samples. Model-based techniques capture the variations in writing from many samples of a desired unit into models.

Concatenation techniques can be subdivided, according to the connection means they adopt, into no-connection, direct-connection, and modeled-connection. No-connection techniques juxtapose writing units into text lines. Direct-connection techniques take writing units and position them such that the ending ligature from one unit (also referred to as tail [21,61,62] or prefix segment [42]) directly connects to the starting ligature of the next unit (also referred to as head or suffix segment) to form a text line. Modeled-connection techniques add new connection ligatures synthesized by parametric curves. The tail segment of a letter and the head segment of the subsequent letter can form control points to the synthesis of a connecting ligature.

In several scripts, there are parts that connect and others that do not. Connecting parts usually use direct-connection and modeled-connection techniques, whereas the non-connecting parts would use no-connection technique.

Table 3 classifies common shape-simulation works, with the type of data and scripting units used. For character synthesis, generation techniques are more popular although concatenation was used to synthesize characters from sub-characters [23,42]. On the other hand, cursive sub-words are mainly concatenated except when they are part of complete lines which are generated using perturbation [13]. For text line synthesis, both concatenation as well as generation techniques are commonly used although no work is reported on online synthesis of text lines using generation techniques. In Sects. 3.1 and 3.2, we discuss generation and concatenation techniques, respectively.

3.1 Generation techniques

As mentioned before, generation can be performed by perturbation, fusion, or modeling, requiring one, two, or more input samples, respectively. Except for perturbation-based techniques, the two other techniques require shape-matching operations [14,50]. Table 4 presents different works classified by the three generation-based techniques along with the

Table 3 Shape-simulation techniques with input/output units and scripting systems

Granularity/technique data type	Character		(Sub-)word		Text lines	
	Concatenation	Generation	Concatenation	Generation	Concatenation	Generation
Online	Rao [42]	Stettiner and Chazan [48]	Rao [42]		Guyon [20]	
	Jawahar and Balasubramanian [23]	Wang et al. [1,21,39]	Wang et al. [1]		Lin et al. [53]	
		Choi et al. [22,47]	Saabni [4,12]		Jawahar and Balasubramanian [23]	
		Zheng and Doermann [14]				
Offline		Mori et al. [41]	Helmers and Bunke [57]		Elarian et al. [38]	Varga and Bunke [13,17,29]
		Viswanath et al. [18,50,51]	Thomas and Govindaraju [5,43]			Vincent et al. [9]
		Miyao [15,36]	Elarian et al. [38]			Cheng and Lopresti [52]
		Thomas and Govindaraju [5,43]				Chen et al. [46]
Mixed		Dolinsky and Takagi [60][a]				Xu [49][c]
		Liu et al. [27][b]				

[a] Printed and online inputs, offline output
[b] Online input and offline-like output
[c] Not specified

Table 4 Generation-based synthesis techniques with data types

Technique data type	Perturbation based	Fusion based	Model based
Online	Lin and Wan [28]	Zheng and Doermann [14]	Rao [42]
	Wang et al. [21]		Stettiner and Chazan [48]
			Choi et al. [47]
			Wang et al. [1,21]
Offline	Varga and Bunke [13,17,29]	Viswanath et al. [18,50,51]	Mori et al. [41]
	Cheng [52] and Chen [46] with Lopresti		Vincent et al. [9]
			Dolinsky and Takagi [40]
Mixed	—	—	Liu et al. [27][a]
			Miyao [15,36][b]

[a] Online and offline
[b] Online inputs–offline outputs

various output data types used. In the following subsections, each of the three generation techniques is discussed in detail.

3.1.1 Perturbation-based generation

Perturbation-based techniques generate new samples by altering geometric features such as the size, thickness, and slant of a given sample. Perturbation-based operations can be seen as the inverse of the preprocessing steps employed in text recognition. Perturbation-based techniques are easy to apply, but the results may be unnatural due to random and non-calibrated parameter settings [29,46,52].

Stroke-wise rotation and scaling perturbations are applied to online strokes with high curvature points in [28]. Perturbations are added to text lines in [21] in order to generate additional training data to increase the variability within the dataset. Varga and Bunke [13,17,29] apply nonlinear geometric perturbations on complete text lines and connected components of offline images. They choose the parameters of their perturbation models randomly from predefined ranges. Their results show that this approach can be useful in improving OCR recognition performance by adding synthesized data to otherwise small training sets. Cheng and Lopresti [52] calibrate the parameters of the perturbation-based model of the work of Varga and Bunke [13]. Chen et al. use those perturbation models for writer identification on Arabic handwritten data [46].

3.1.2 Fusion-based generation

Fusion-based techniques take few input samples and combine them into new synthesized outputs. They differ from concatenation techniques in that they generate scripting units at the same level as their inputs; e.g., characters generate new characters. Shape-matching algorithms are necessary for fusion-based techniques to make sure that segments are properly aligned. The number of unique outputs is limited in fusion-based techniques as compared to that of other generation techniques.

Zheng et al. [14] present a point-matching algorithm and apply it to generate online Latin letters by displacing the points in the range between two samples. Viswanath et al. [18,50,51] implicitly combine different partitions of samples of offline images into hybrid images while fixing their shared components. Fusion-based handwriting synthesis is not very common in the literature, probably because it is not as established as model-based techniques.

3.1.3 Model-based generation

Model-based techniques capture the statistics of natural handwriting variations into models. Although model-based techniques are profoundly established in theory, they may often be challenging to implement due to the large number of samples they require [14]. Models resulting from these techniques can also be utilized in recognition systems [42,63].

Model-based generation may process sampled points of data often chosen for their structural features, e.g., maximum curvature [21] or zero-velocity [48], by spatial sampling, e.g., equidistance [40] or by drawing them from a generative statistical recognizer, e.g., a Bayesian network [47]. A common modeling scenario is that statistics on displacements of the sample points from a template sample are captured. New sample points are then drawn from the statistical model to generate shapes.

Techniques adopted for model-based generation depend, again, on the target applications and data types. In the following, we discuss the various techniques for model-based generation under online and offline categories.

Techniques that use online data. Different techniques are used to sample the drawn coordinates of online data. One can extract straight graphemes within online characters and select them to be control points [42]. From these control points, more significant ones can be selected using Gabor filters [21] or Principle Component Analysis (PCA) [1]. Some works avoid the sampling of points and generate the coordinates directly [47,48].

Once control points are selected from the online data, characters can be synthesized by using polynomial splines by connecting the control points [42]. One approach [21] is to match the control points to a template that is computed from all the sample characters. Synthesis is done by drawing the control points according to a generative model of their displacements from the template and then using curves (splines) to connect them into a character shape. Some authors have used eigen vectors instead of splines [1].

Techniques that do not directly rely on the extraction of control points from sample characters define generative models from which new samples can be synthesized. Some authors use generative statistical systems to synthesize handwriting through sampling from estimated joint distributions [47]. Others consider the online x and y sequences of single-stroke character shapes as the impulse response of an online signal [48]. The authors sample characters into fixed sized vectors and match the points by using the modified Newton method. They find the character synthesizing filters by solving the optimization problems of the transfer functions for each pair of inputs and matched outputs.

Techniques that use offline data. These techniques work on the images of handwritten texts. A natural idea is to derive some template patterns from the offline data and then generate new samples from the templates. In [41], all the points from a sample of training data are matched with its class template (taken as the average of all available samples for that class) and their displacements are recorded. Then, generation of new samples is done by selecting new points within the pre-calculated displacements. A similar approach of generating samples from templates using the displacements is used in [40]. However, the authors used characters from standard fonts as templates. To calculate the displacements, the outlines of font templates are sampled equidistantly to match it with the offline images.

In another approach, Vincent et al. [9] applied fractal decomposition and synthesis as a lossy encoding–decoding process to offline character images. They defined reference bases that are repeated in an alphabet and then used these to model characters of the alphabet.

Techniques that use mixed online and offline data. There are several works that try to take benefits of both online and offline data. In [36], online strokes help in defining affine perturbations to offline data. Online samples of some Hiragana characters were matched to a selected template sample by dynamic programming. Using PCA on the differences between the template and the samples, the patterns with most effective impact were chosen to improve the variety of patterns in [15].

Liu et al. [27] patented the idea of using trained Hidden Markov Models (HMMs) as generative statistical models to synthesize handwritten samples. The HMMs were trained as handwriting recognizers using handwritten and calligraphic-font samples. Pressure and ink data provided online and offline flavored outputs.

3.2 Concatenation techniques

Concatenation refers to any synthesis approach that combines input samples into outputs of higher semantic levels. One common example is the concatenation of character shapes into words or text lines. Concatenation can be seen as the reverse of character segmentation in a text recognition system. It encompasses tasks such as baseline detection, horizontal space modeling, connection part segmentation and modeling, and segment joining and trimming. The input units for concatenation techniques are usually characters [53] but can also be sub-characters [42], character groups [20], or connected components [38].

Concatenation techniques depend on the knowledge of the rules of a writing script. Some scripts, such as Arabic, enforce most characters to be joined in a continuous flow [64], whereas the composite style of Latin allows the writer to connect or disconnect letters, and the Chinese script does not connect characters together but is often composed of interconnected components/strokes.

The shape of the segments connecting letters, referred to as ligatures in [5], also depends on the script. In Latin, they often ascend in a curvy line to connect the suffix segment of a letter to the prefix segment of the subsequent letter [42]. The Arabic connection (Kashidah) is usually horizontal with occasional vertical ligatures [38].

Table 5 summarizes the different works for online and offline no-connection, direct-connection, and modeled-connection categories.

Table 5 Classification of works according to concatenation techniques and data types

Concatenation technique data type	No-connection	Direct-connection	Modeled-connection
Online	Guyon [20][a]	Saabni [12]	Rao [42]
	Jawahar and Balasubramanian [23][b]		Wang et al. [1,21]
			Lin and Wan [28]
Offline	Elarian et al. [38]	Elarian et al. [38]	Thomas et al. [5]

[a] Sequential juxtaposition. Inking effect is added in one of the variations to generate an offline-like version of this approach
[b] Overlapping juxtaposition

3.2.1 No-connection concatenation

No-connection techniques concatenate scripting units by aligning and juxtaposing them without connection. Many techniques can be employed for the alignment of the text components such as projection profiles, baseline estimation, and connected component analysis. Usually, a number of techniques are used together for robust alignment [38,43].

Guyon [20] suggests simple juxtaposition of selected letter strings to synthesize semi-cursive text. Letter groups are selected based on their frequency in a linguistic corpus. In the training phase, a sample of each of the letter strings is collected from the writer whose handwriting is to be imitated on an online tablet. In the synthesis phase, the text to be synthesized is parsed into a sequence of available letter strings and the corresponding letter string images are placed as text lines and paragraphs. This approach works well in subjective tests at the first glance. However, the trained eye may soon notice abrupt pen lifts between glyphs, repetitions of glyph appearance, and too regular pressure or inking. Geometric transformations are introduced to reduce such effects. In [38], non-connecting PAWs (Parts of Arabic Words) are aligned without any connection.

3.2.2 Direct-connection concatenation

Direct-connection techniques take writing units and position them such that the ending ligature from one unit directly connects to the starting ligature of the next unit to form text lines. These techniques are suitable for inherently cursive scripts like Arabic. Arabic online handwritten samples have been segmented and later concatenated to produce new samples in [12]. Similar ideas for segmenting, sampling, and concatenating Latin letters were proposed in [1,21,28] and patented in [53]. In [38], samples of offline Arabic segmented letters are conditionally selected and later connected directly using the horizontal connection stroke (Kashidah).

3.2.3 Modeled-connection concatenation

Modeled-connection techniques add new connection ligatures synthesized by parametric curves. Rao [42] modeled the connection between the suffix segment of a letter to the prefix segment of the subsequent letter using polynomial and Bezier curves. His results of letter to letter concatenation are reported to appear natural, provided the segments of letters are adequately extracted.

Wang et al. [21] and Xu et al. [49] developed a letter concatenation model in addition to a letter generation model. Their letter concatenation technique is similar to that of [42]: They concatenate the tail segment of a letter to the head segment of the subsequent letter (corresponding to the suffix and prefix segment in Rao's work, respectively) to minimize energy in a deformable model.

Style preserving concatenation [28] suggests connecting Latin letters according to some probabilities that reflect the writer's style. Whenever it is decided that letters should be connected, the extensions (probably trimmed) are connected with interpolation. If it is decided that letters should not be connected, an ending-position, rather than a middle position, sample of the letter is used (i.e., a no-connection technique).

Cursive handwritten CAPTCHAs are produced by the concatenation of skeletonized letters at the level of the baseline [43]. They define their connection ligatures by looking at the derivative of the vertical projection. They parameterize ligatures and join them from the end of a letter to the body of the next letter. Table 6 summarizes some key shape-simulation works.

4 Overview of some other synthesis approaches

In this section, we present techniques for handwriting synthesis which are non-shape simulation approaches. *Movement-simulation* approaches are the most common non-shape simulation techniques. Movement simulation is a top-down approach to handwriting synthesis where the neuromuscular acts of writing are simulated. One approach to synthesizing handwritten data is to model strokes as oscillatory components, where the letter formation is a result of horizontal and vertical oscillations (i.e., constrained modulation). The horizontal oscillation and its modulation control the stroke/letter shape, and the vertical oscillation and its modulation control the letter height [62]. Motivated by this idea, Gangadhar et al. proposed a neural network model of handwriting strokes, where the stroke velocities are expressed as oscillatory neural activities. The architecture has stroke selection as the input layer and the estimated stroke velocities are represented by the output layer [3].

One of the most notable contributions for modeling strokes is by Plamondon and his group [6–8,55,65–67]. The strokes are defined from the context of kinematic theory of rapid human movement as primitive movement units which can be superimposed to construct word patterns [16]. A stroke model describes the essential characteristics of the pen-tip trajectory [68]. The main idea behind the kinematic theory is that a neuromuscular systems involved in the production of a rapid movement can be considered as a linear system made up of a large number of coupled subsystems, and the impulse response of such system converges toward a lognormal function under certain conditions [66,67,69]. There are many models derived from this lognormal paradigm. These models can be broadly categorized into two: (i) Delta-lognormal model, which involves two neuromuscular systems (each described by a lognormal impulse response and timing properties), one agonist to, and the other antagonist to, the direction of the movement. This model generates straight strokes and predicts all the velocity patterns observable in a set of strokes; and (ii) sigma-lognormal model, where the assumption is that the two neuromuscular systems do not work in exactly opposite directions, and thus, the resultant velocity is described by the vectorial summation of the contribution of each of the neuromuscular systems involved.

All the different models differ in their stroke generation quality depending on the number of parameters used in a given model (the simple one with three parameters to the more complex ones having up to 11 parameters) [55]. Estimating the parameters robustly is one of the issues in using these stroke models for handwriting synthesis. Moreover, the variability of handwriting, as a result of varying the parameter values, to generate realistic text needs further investigation. There are many methods proposed to estimate the initial parameters of the lognormal stroke models like INFLEX, INITRI, and the XZERO [68,70,71]. Each of the algorithms has its advantages and limitations, and the authors have proposed using hybrid versions of them as they seem complimentary to each other.

In [16], the authors presented a system for synthesizing a large database of handwriting from few specimens using the sigma-lognormal model. The system can be used to generate writing samples for a single writer, as well as in multi-writer setup. The variability observed in handwriting data can be regenerated by varying the sigma-lognormal parameters around their mean values within the limits fixed by their standard deviations. The factor of variability needs to be carefully fixed so as to get intelligible samples.

In another approach, time trajectories of the English alphabet were modeled using an oversampled reverse time delay neural network (TDNN) architecture to generate outputs that can control the writing of characters with a pen [72]. The network was trained on character glyphs as a sequence of successive points in time. Three outputs provided the time sequences of signals that controlled the X and Y positions of the pen and up/down pen control.

Bayoudh et al. [10] propose using the principle of analogical proportion to synthesize new examples from an existing limited set of real examples. Each character is represented

Table 6 Summary of the specification of shape-simulation systems

Author(s) (citations)	Technique	Input granularity	Output granularity	Online/offline	Applications	Evaluation method	Script
Rao [42]	Concatenation	Characters	Cursive writing	Online	OCR data compression	Subjective and analysis by synthesis	Latin
	Model based	Sub-characters	Characters				
Stettiner and Chazan [48]	Model based	Character	Character	Online	OCR, Writer-imitation and identification	Test of completeness for the model	Latin
Guyon [20]	Concatenation	Glyphs	Semi-cursive writing	Online + inking	Personalization, pleasant view	Subjective	Latin
Mori et al. [41]	Model based	Digits	Digits	Offline	OCR	–	Digits
Wang et al. [21]	Concatenation	Ligatures	Cursive writing	Online	Pen-based computers	Subjective	Latin
	Model + some perturbation	Characters	Characters				
Wang et al. [1]	Concatenation + sampling	Characters with extensions	Cursive writing	Online	Pen-based computers	–	Latin
	Model based	Segmented samples	Characters				
Helmers and Bunke [57]	Concatenation	1. Isolated characters	Cursive writing	Offline	OCR	OCR	Latin
		2. Characters from text					
		3. n-tuples of characters					
Choi et al. [22,47]	Model based	Characters	Characters	Online	Personalization	Subjective	Hangul and digits
Varga and Bunke [13,17,29]	Perturbation	1. Text line	1. Text line	Offline	Training data for HMM-based OCR	OCR training	Latin
		2. Connected components	2. connected components				
Viswanath et al. [18,50,51]	Fusion based	Characters	Characters	Offline	Nearest neighbor classifier	Nearest neighbor classifier	Digits
Zheng and Doermann [14]	Fusion based	Characters	Characters	Online	OCR, writer identification	Deformation error	Latin
Vincent et al. [9]	Model based	Text line	Text line	Offline	Scripting system recognition, compression	Language and script recognition	Latin
							Arabic
							Cyrillic
Miyao [15,36]	Model based	Online characters	New offline characters	Online	Offline OCR	OCR training	Hiragana
Dolinsky and Takagi [40,44,60]	Model based	1. Font character	Characters	Both	Human-like behavior, personalized PC	Errors of recurrent neural networks	Hiragana
		2. Handwritten samples					

Table 6 continued

Author(s) (citations)	Technique	Input granularity	Output granularity	Online/offline	Applications	Evaluation method	Script
Lin and Wan [28]	Concatenation + sampling	Characters	Cursive writing	Online	Aesthetical and personal view, forensics, for disabled, OCR, captchas	Subjective	Latin
	Perturbation	Characters	Characters				
Thomas and Govindaraju [5,43]	Concatenation	Characters	Cursive writing	Offline	CAPTCHA	Subjects and OCR performances	Latin
	Perturbation	Characters	Characters				
Elarian [38]	Concatenation + sampling	Segmented characters	Text lines	Offline	OCR, word spotting	Comparison between *best* and *worst* synthesis	Arabic
Saabni and El-Sanaa [4,12]	Concatenation	Segmented characters	Connected components	Online	Holistic OCR	OCR training	Arabic
Cheng and Lopresti [52]	Perturbation	Text line	Text line	Offline	OCR	Subjective	Latin
Chen et al. [46]	Perturbation	Text line	Text line	Offline	Writer identification	Writer identification	Arabic
Jawahar and Balasubramanian [23]	Model + Concatenation	Characters	Characters and words	Online	OCR, personalization, study of human style	Subjective	Indian (Hindi, Tamil, Malayalam, Telugu)
Fujioka [37]	Generation and concatenation	Character	Cursive writing	Offline	Calligraphy	Subjective	Japanese (Kana)
Xu [49]	Generation and concatenation	Text	Novel cursive text	Unspecified	Personalization	NA (patent)	Latin
Liu et al. [27]	Model	One or more characters	Characters probably with inking	Online → offline-like	Personalization and artistic view	NA (patent)	Latin
Lin et al. [53]	Concatenation	Character positions	Text line	Online	Training OCR	NA (patent)	Latin

as a sequence of Freeman chain codes including a set of anchorage points. Experiments evaluated the improvement in the training of a set of classifiers on character recognition rate as a result of increasing the size of the dataset. The results confirmed that the proposed approach is as effective as character synthesis through knowledge-based approaches in the form of image-based (scant and slat) distortions and online (speed and curvature) distortions.

Slim and Benrejeb [73] modeled the handwriting process of few Arabic letters using electromyographic signals (EMG) generated by muscles in the forearm. An RBF neural network with feedback and time delay learns to associate the EMG signals generated, as a character is drawn, with the sequence of pen displacements recorded in the X and Y directions. Inverse models are also described for generating the EMG signals from the recorded position signals.

5 Summary and conclusions

Handwriting synthesis is the artificial generation of images that resemble human writing. Synthesis has several applications such as the improvement of text recognition systems, font personalization, and CAPTCHAs. The most predominant applications of synthesis are those related to recognition. The most common approaches to handwriting synthesis are shape simulation and movement simulation. Shape-simulation approaches for handwriting synthesis model handwritten shapes, while movement-simulation approaches model neuromuscular movements. Important aspects of handwriting synthesis, such as input/output levels, data types, and script, were discussed. Different evaluation methods were reported along with the applications that they suit. We noticed that subjective evaluation methods are used more frequently in the literature than objective ones.

The works on shape-simulation generation and concatenation techniques were classified and surveyed. The most common unit for handwriting generation is the character. Subwords are the most prevailing output units for handwriting concatenation. We noticed that among the techniques of synthesis, fusion-based generation is probably the most underexplored. Some important non-shape simulation techniques were also presented.

Some challenges and possibilities for future works were identified: Offline handwriting lacks temporal information which makes the identification of its segments more challenging as compared to online handwriting. Besides, offline data have inking effects which need special treatment when processed. On the other hand, the synthesis of online data requires velocity and pressure estimation. Moreover, there is a lack of standard benchmarks for objective comparisons of different techniques. In addition, the challenging problem of segmentation is necessitated by some techniques. Not much work has been reported addressing the issue of sample selection of consequent units for concatenation.

Acknowledgments The authors would like to thank King Fahd University of Petroleum & Minerals (KFUPM), Qassim University, in addition to King Abdul Aziz City for Science and Technology (KACST) for supporting this work under the grant "GSP-18-112." The authors would also like to thank the anonymous reviewers for their valuable comments which greatly improved the presentation of this paper.

References

1. Wang, J., Wu, C., Xu, Y.-Q., Xu, Y., Shum, H.: Combining shape and physical models for on-line cursive handwriting synthesis Int. J. Doc. Anal. Recognit. **7**(4), 219–227 (2005)
2. Chang, W.-D., Shin, J.: A statistical handwriting model for style-preserving and variable character synthesis. Int. J. Doc. Anal. Recognit. IJDAR **15**(1), 1–19 (2012)
3. Gangadhar, G., Joseph, D., Chakravarthy, V.S.: An oscillatory neuromotor model of handwriting generation. Int. J. Doc. Anal. Recognit. IJDAR **10**(2), 69–84 (2007)
4. Saabni, R.M., El-Sana, J.A.: Comprehensive synthetic Arabic database for on/off-line script recognition research. Int. J. Doc. Anal. Recognit. IJDAR **16**(3), 285–294 (2013)
5. Thomas, A.O., Rusu, A., Govindaraju, V.: Synthetic handwritten CAPTCHAs. Pattern Recognit. **42**(12), 3365–3373 (2009)
6. Galbally, J., Fierrez, J., Ortega-Garcia, J., Plamondon, R.: Synthetic on-line signature generation. Part II: experimental validation. Pattern Recognit. **45**(7), 2622–2632 (2012)
7. Galbally, J., Plamondon, R., Fierrez, J., Ortega-Garcia, J.: Synthetic on-line signature generation. Part I: methodology and algorithms. Pattern Recognit. **45**(7), 2610–2621 (2012)
8. O'Reilly, C., Plamondon, R.: Development of a sigma-lognormal representation for on-line signatures. Pattern Recognit. **42**(12), 3324–3337 (2009)
9. Vincent, N., Seropian, A., Stamon, G.: Synthesis for handwriting analysis. Pattern Recognit. Lett. **26**(3), 267–275 (2005)
10. Bayoudh, S., Mouchère, H., Miclet, L., Anquetil, E.: Learning a classifier with very few examples: analogy based and knowledge based generation of new examples for character recognition. In: Kok, J.N., Koronacki, J., de Mantaras, R.L., Matwin, S., Mladenič, D., Skowron, A. (eds.) Machine Learning: ECML 2007, pp. 527–534. Springer, Berlin (2007)
11. Jin, L., Zu, X.N.: Synthesis of Chinese character using affine transformation. In: Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007, vol. 1, pp. 218–222.(2007)
12. Saabni, R., El-Sana, J.: Efficient generation of comprehensive database for online arabic script recognition, In: 10th International Conference on Document Analysis and Recognition, 2009. ICDAR '09, pp. 1231–1235 (2009)
13. Varga, T., Bunke, H.: Generation of synthetic training data for an HMM-based handwriting recognition system. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), pp. 618–622 (2003)
14. Zheng, Y., Doermann, D.: Handwriting matching and its application to handwriting synthesis. In: Proceedings of ICDAR, pp. 861–865 (2005)
15. Miyao, H., Maruyama, M.: Virtual example synthesis based on PCA for off-line handwritten character recognition. In: Proceedings of the 7th International Conference on Document Analysis Systems. Berlin, Heidelberg , pp. 96–105 (2006)
16. Djioua, M., Plamondon, R.: An interactive system for the automatic generation of huge handwriting databases from a few specimens. In: 19th International Conference on Pattern Recognition, 2008. ICPR 2008, pp. 1–4 (2008)
17. Varga, T., Bunke, H.: Off-line handwritten textline recognition using a mixture of natural and synthetic training data. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, pp. 545–549 (2004)
18. Viswanath, P., Murty, M.N., Bhatnagar, S.: A pattern synthesis technique with an efficient nearest neighbor classifier for binary pattern recognition. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4–Volume 04, Washington, DC, USA, pp. 416–419 (2004)
19. Zi, G., Doermann, D.: Document image ground truth generation from electronic text. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, pp. 663–666 (2004)
20. Guyon, I.: Handwriting synthesis from handwritten glyphs. In: Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition, pp. 309–312 (1996)
21. Wang, J., Wu, C., Xu, Y.-Q., Shum, H.-Y., Ji, L.: Learning-based cursive handwriting synthesis, In: Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 157–162 (2002)
22. Choi, H., Choi, S.J., Kim, J.-H.: Writer dependent online handwriting generation with Bayesian network. In: Ninth International Workshop on Frontiers in Handwriting Recognition, pp. 130–135 (2004)
23. Jawahar, C.V., Balasubramanian, A.: Synthesis of online handwriting in Indian languages. In: The Tenth International Workshop on Frontiers in Handwriting Recognition (2006)
24. Ballard, L., Lopresti, D., Monrose, F.: Evaluating the security of handwriting biometrics. In: The 10th International Workshop on the Foundations of Handwriting Recognition, pp. 461–466 (2006)
25. Kokula, M.: Automatic generation of script font ligatures based on curve smoothness optimization Electron. Publ. **7**(4), 217–229 (1994)
26. Sarfraz, M., Khan, M.A.: Automatic outline capture of Arabic fonts. Inf. Sci. **140**(3–4), 269–281 (2002)
27. Liu, P., Ma, L., Soong, F.K.-P., Wu, Y.-J.: Hidden markov model based handwriting/calligraphy generation. WO2009023648 A2 (2009)
28. Lin, Z., Wan, L.: Style-preserving English handwriting synthesis. Pattern Recognit. **40**(7), 2097–2109 (2007)
29. Varga, T., Bunke, H.: Perturbation models for generating synthetic training data in handwriting recognition. In: Marinai, P.S., Fuji-

sawa, P.H. (eds.) Machine Learning in Document Analysis and Recognition, pp. 333–360. Springer, Berlin (2008)
30. Srihari, S.N., Srinivasan, H., Huang, C., Shetty, S.: Spotting words in Latin, Devanagari and Arabic scripts. Vivek Indian J. Artif. Intell. **16**(3), 2–9 (2006)
31. Serrano, J.A.R., Perronnin, F.C.: Handwritten word spotter using synthesized typed queries. US20100067793 A1 (2010)
32. Coates, A.L., Baird, H.S., Faternan, R.J.: Pessimal print: a reverse Turing test. In: Sixth International Conference on Document Analysis and Recognition, pp. 1154–1158 (2001)
33. Gutub, A., Ghouti, L., Elarian, Y., Awaideh, S., Alvi, A.: Utilizing diacritic marks for arabic text steganography. Kuwait. J. Sci. Eng, **37**(1), 89–109 (2010)
34. Odeh, A., Elleithy, K., Faezipour, M.: Steganography in arabic text using Kashida variation algorithm (KVA). In: Systems, Applications and Technology Conference (LISAT), 2013 IEEE Long Island, pp. 1–6 (2013)
35. Nel, E.-M., du Preez, J.A., Herbst, B.M.: Verification of dynamic curves extracted from static handwritten scripts. Pattern Recognit. **41**(12), 3773–3785 (2008)
36. Miyao, H., Maruyama, M., Nakano, Y., Hananoi, T.: Off-line handwritten character recognition by SVM based on the virtual examples synthesized from on-line characters. In: Eighth International Conference on Document Analysis and Recognition, 2005. Proceedings, vol. 1, pp. 494–498 (2005)
37. Fujioka, H., Kano, H., Nakata, H., Shinoda, H.: "Constructing and reconstructing characters, words, and sentences by synthesizing writing motions. IEEE Trans. Syst. Man Cybern. Part Syst. Hum. **36**(4), 661–670 (2006)
38. Elarian, Y., Ahmad, I., Awaida, S., Al-Khatib, W.G., Zidouri, A.: An arabic handwriting synthesis system. Pattern Recognit. (2014). doi:10.1016/j.patcog.2014.09.013
39. Wang, Y., Wang, H., Pan, C., Fang, L.: Style preserving Chinese character synthesis based on hierarchical representation of character. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, pp. 1097–1100 (2008)
40. Dolinsky, J., Takagi, H.: Synthesizing handwritten characters using naturalness learning. In: IEEE International Conference on Computational Cybernetics, 2007. ICCC 2007, pp. 101–106 (2007)
41. Mori, M., Suzuki, A., Shio, A., Ohtsuka, S.: Generating new samples from handwritten numerals based on point correspondence. In: 7th International Workshop on Frontiers in Handwriting Recognition, pp. 281–290 (2000)
42. Rao, P.: Shape vectors: an efficient parametric representation for the synthesis and recognition of hand script characters. Sadhana **18**(1), 1–15 (1993)
43. Thomas, A., Rusu, A., Govindaraju, V: Generation and performance evaluation of synthetic handwritten CAPTCHAs. J. Pattern Recognit. **42**, 3365–3373 (2009) (Special Issue on Handwriting Recognition)
44. Dolinsky, J.: Naturalness Learning and Its Application to the Synthesis of Handwritten Characters (Soft Computing, < Special Issue > Doctorial Theses on Artifical Intelligence), vol. 25, (2010)
45. Ballard, L., Monrose, F., Lopresti, D.: Biometric authentication revisited: understanding the impact of wolves in sheep's clothing. In: Proceedings of the 15th Conference on USENIX Security Symposium—Volume 15, Berkeley, CA, USA (2006)
46. Chen, J., Cheng, W., Lopresti, D.: Using perturbed handwriting to support writer identification in the presence of severe data constraints In: 18th Document Recognition and Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium (2011)
47. Choi, H., Cho, S.-J., Kim, J.H.: Generation of handwritten characters with Bayesian network based on-line handwriting recognizers. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, vol. 2, p. 995. Washington, DC, USA (2003)
48. Stettiner, O., Chazan, D.: A statistical parametric model for recognition and synthesis of handwriting. In: Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing, vol. 2, pp. 34–38 (1994)
49. Xu, Y.-Q., Shum, H.-Y., Wang, J., Wu, C.: Learning-based system and process for synthesizing cursive handwriting. US7227993 B2 (2007)
50. Viswanath, P., Murty, N., Bhatnagar, S.: Overlap pattern synthesis with an efficient nearest neighbor classifier. Pattern Recognit. **38**(8), 1187–1195 (2005)
51. Viswanath, P., Murty, M.N., Bhatnagar, S.: Partition based pattern synthesis technique with efficient algorithms for nearest neighbor classification. Pattern Recognit. Lett. **27**(14), 1714–1724 (2006)
52. Cheng, W., Lopresti, D.: Parameter calibration for synthesizing realistic-looking variability in offline handwriting. In: 18th Document Recognition and Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium, vol. 7874 (2011)
53. Lin, Z., Wan, L., Hu, C.-H., Wang, J.: Handwriting recognition training and synthesis. US7657094 B2 (2010)
54. Margner, V., Pechwitz, M.: Synthetic data for Arabic OCR system development, In: Sixth International Conference on Document Analysis and Recognition, 2001. Proceedings, pp. 1159–1163 (2001)
55. Plamondon, R., Djioua, M.: A multi-level representation paradigm for handwriting stroke generation. Hum. Mov. Sci. **25**(4–5), 586–607 (2006)
56. Ha, T.M., Bunke, H.: Off-line handwritten numeral string recognition. Pattern Recognit. **31**, 257–272 (1997)
57. Helmers, M., Bunke, H.: Generation and use of synthetic training data in cursive handwriting recognition. In: Perales, F.J., Campilho, A.J.C., de la Blanca, N.P., Sanfeliu, A. (eds.) Pattern Recognition and Image Analysis, pp. 336–345. Springer, Berlin (2003)
58. Chelba, C., Bikel, D., Shugrina, M., Nguyen, P., Kumar, S.: Large scale language modeling in automatic speech recognition. CoRR, vol. abs/1210.8440 (2012)
59. Chun-Hui, H., Zhouchen, L., Liang, W., Jian, W.: Handwriting recognition training and synthesis. US20070154094 A1 (2007)
60. Dolinsky, J., Takagi, H.: Analysis and modeling of naturalness in handwritten characters. IEEE Trans. Neural Netw. **20**(10), 1540–1553 (2009)
61. Bezine, H., Alimi, A.M., Derbel, N.: Handwriting trajectory movements controlled by a beta-elliptical model. In: Proceedings of Seventh International Conference Document Analysis and Recognition, pp. 1228–1232 (2003)
62. Hollerbach, J.M.: An oscillation theory of handwriting. Biol. Cybern. **39**(2), 139–156 (1981)
63. Elbert, B.L.: Analysis by synthesis in handwriting recognition. Thesis, Massachusetts Institute of Technology (1997)
64. "Cursive", Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Cursive
65. Djioua, M., Plamondon, R.: Studying the variability of handwriting patterns using the kinematic theory. Hum. Mov. Sci. **28**(5), 588–601 (2009)
66. Plamondon, R.: A kinematic theory of rapid human movements. Biol. Cybern. **72**(4), 309–320 (1995)
67. Plamondon, R.: A kinematic theory of rapid human movements: part III. Kinetic outcomes. Biol. Cybern. **78**(2), 133–145 (1998)
68. Plamondon, R., Li, X., Djioua, M.: Extraction of delta-lognormal parameters from handwriting strokes. Front. Comput. Sci. China **1**(1), 106–113 (2007)
69. Plamondon, R., Feng, C., Woch, A.: A kinematic theory of rapid human movement. Part IV: a formal mathematical proof and new insights. Biol. Cybern. **89**(2), 126–138 (2003)

70. Djioua, M., Plamondon, R.: A new algorithm and system for the characterization of handwriting strokes with delta-lognormal parameters. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 2060–2072 (2009)
71. Guerfali, W., Plamondon, R.: The delta lognormal theory for the generation and modeling of cursive characters. In: Proceedings of the Third International Conference on Document Analysis and Recognition, 1995, vol. 1, pp. 495–498 (1995)
72. Simard, P., Cun, Y.L.: Reverse TDNN: an architecture for trajectory generation. In: Advances in Neural Information Processing Systems, vol. 4, pp. 579–588 (1991)
73. Slim, M.A., Abdelkarim, A., Benrejeb, M.: Handwriting process modelling by artificial neural networks. Int. J. Comput. Inf. Syst. Ind. Manag. Appl. **5**, 297–307 (2013)