Pattern Anal Applic (2017) 20:307–323
DOI 10.1007/s10044-017-0607-x

SURVEY

Segmentation and recognition system for unknown-length handwritten digit strings

Abdeljalil Gattal[1,2] · Youcef Chibani[3] · Bilal Hadjadji[3]

Received: 14 January 2016 / Accepted: 28 January 2017 / Published online: 9 February 2017

Abstract The segmentation of handwritten digit strings into isolated digits remains a challenging task. The difficulty for recognizing handwritten digit strings is related to several factors such as sloping, overlapping, connecting and unknown length of the digit string. Hence, this paper aims to propose a segmentation and recognition system for unknown-length handwritten digit strings by combining several explicit segmentation methods depending on the configuration link between digits. Three segmentation methods are combined based on histogram of the vertical projection, the contour analysis and the sliding window Radon transform. A recognition and verification module based on support vector machine classifiers allows analyzing and deciding the rejection or acceptance each segmented digit image. Moreover, various submodules are included leading to enhance the robustness of the proposed system. Experimental results conducted on the benchmark dataset show that the proposed system is effective for segmenting handwritten digit strings without prior knowledge of their length comparatively to the state of the art.

✉ Abdeljalil Gattal
a_gattal@esi.dz; Ab.gattal@gmail.com;
ab.gattal@univ-tebessa.dz

Youcef Chibani
ychibani@usthb.dz

Bilal Hadjadji
bhadjadji@usthb.dz

1 Ecole Nationale Supérieure d'Informatique (ESI), BP 68M, 16309 OUED SMAR, El-Harrach, Algiers, Algeria

2 LAMIS Laboratory, Larbi Tebessi University, Route de Constantine, 12002 Tébessa, Algeria

3 LISIC Lab. , Faculty of Electronics and Computer Science, University of Science and Technology Houari Boumédienne (USTHB), PO Box 32, 16111 Bab-Ezzouar, El-Alia, Algiers, Algeria

Keywords Histogram projection · Contour analysis · Sliding window · Radon transform · Digit recognition–verification · SVM

1 Introduction

The Automatic Handwritten Digit String Recognition (AHDSR) is required in many applications such as the amount of the bank checks [1–3], postal code [4] and forms [5]. In this context, two main problems occurred when attempting to design a handwritten digit string recognition system. The first problem is the unknown length of the digit string, which is not carefully written by people in real-life situations [6]. The second problem is the link between adjacent digits, which can be naturally spaced, overlapped or/and connected [7].

Hence, various AHDSR systems have been developed in the past years, which can be divided into two approaches: implicit and explicit. The implicit approach considers all traced points as potential segmentation points [8]. In this case, the segmentation and recognition are performed simultaneously for recognizing a digit string. Indeed, this approach does not attempt to separate digits, but rather it incorporates the implicit segmentation into the recognition module. For instance, Britto et al. [6] proposed an AHDSR based on the hidden Markov models (HMM). In their approach, the segmentation–recognition is performed in two steps. The first one takes into account some contextual information to deliver several segmentation–recognition hypotheses for a given preprocessed string. These hypotheses are used in a second step by using an isolated digit classifier for verifying the accuracy of the recognition. The performance evaluation has been conducted by considering the known-length strings and the unknown-length strings using the length predictor.

The explicit approach is, in contrast, performed by finding the best way to isolate adjacent digits before recognition. In this case, the segmentation and recognition are performed separately. Also, three cases can be occurred when attempting to separate two adjacent digits: spaced, overlapped and/or connected digits. In this case, the design of a robust handwritten digit string recognition system can be performed in two steps: primary and secondary segmentation. The primary segmentation is performed when the adjacent digits are naturally spaced. In this case, digits are separated using a simple histogram of the vertical projection [9]. This segmentation requires a correct writing of the digit image to avoid separating a digit into multiple fragments. In contrast, the secondary segmentation is performed on the resulting image produced from the primary segmentation in order to detect any presence of overlapped digits and/or connected digits. Hence, various methods have been developed specifically for overlapped and connected digits.

Segmentation methods dedicated for overlapped digits are performed when digits are slopped or appear to imperfect alignment between the axis and the written text. Under some conditions, it is possible to apply the methods used in cursive script text to estimate the skew value [10]. In some cases, these methods do not provide an efficient segmentation when overlapped digits are complicated. In contrast, connected digits are the frequent situations occurred when attempting to separate two adjacent digits. Hence, some recent segmentation methods dedicated for connected digits have been proposed for single or multiple touching such as the drop-fall-based algorithms [1], the thinning-based algorithms [11], the water reservoir algorithms [12] and the removing useless stroke algorithm [13].

In this context, Oliveira et al. [14] proposed a segmentation and recognition system using some heuristics to avoid an over- or under-segmentation combined with the classifier and the verifier based on multi-layer perceptron (MLP) neural networks. All segmentation points are used to build a segmentation graph. The best segmentation hypothesis is the shortest path of the graph. A post-processing is used to resolve some problems where the verifiers failed and vice versa, and the global decision module makes an accept/reject decision. The performance evaluation of the system has been conducted on NSTRING SD19 database by considering the unknown-length digit strings. Latter, Oliveira and Sabourin [15] proposed to replace the MLP classifier by the support vector machine (SVM) classifier in order to improve the accuracy of the digit string recognition system. Sadri et al. [16] described a genetic framework using contextual knowledge for segmentation and recognition of handwritten digit strings. This method has been developed to locate feature points on the string image in order to generate possible segmentation hypotheses. A genetic algorithm is used as a general search technique for finding the global optimum segmentation hypotheses in digit strings. Experimental results have been conducted on the digit strings extracted from NSTRING SD19 database using two different classifiers (MLP and SVM) by considering the known- and unknown-length digit strings.

We propose, in this paper, a new design of a handwritten digit string recognition system based on the explicit approach for the unknown-length digit strings. Three methods are combined according to the link of adjacent digits, which are the histogram of the vertical projection dedicated for spaced digits, the contour analysis dedicated for overlapped digits and the Radon transform performed on the sliding window dedicated for connected digits.

Our ultimate objective is to propose a full segmentation algorithm based on the explicit approach for handwritten digit string recognition with the unknown-length string.

The paper is organized as follows. In Sect. 2, we describe the used explicit segmentation methods for which we propose a new method for segmenting connected digits by means of the Radon transform. Section 3 is devoted to present the design of the digit string recognition system. Section 4 presents experimental results conducted on the well-known NSTRING SD19. Section 5 concludes the paper and proposes future works.

2 Segmentation methods of the digit string

The choice of digit string segmentation depends on the link between adjacent digits. Hence, four situations occurred which are spaced, overlapped, connected and overlapped/connected digits, respectively. Thus, this section is devoted to describe segmentation methods used in our system for which we propose a new segmentation method performed specifically for connected digits.

2.1 Segmentation of spaced digits

The segmentation of spaced digits is performed using the histogram of the vertical projection (HVP) when digits are well written with some degree of neatness (not connected or not overlapped). The HVP is performed on the binary digit string from which a simple count of the black pixels in each column is running in order to detect the white space between successive digits [9]. This allows determining the location of each component contained in the image. The advantage of this method is its ability to segment the digit

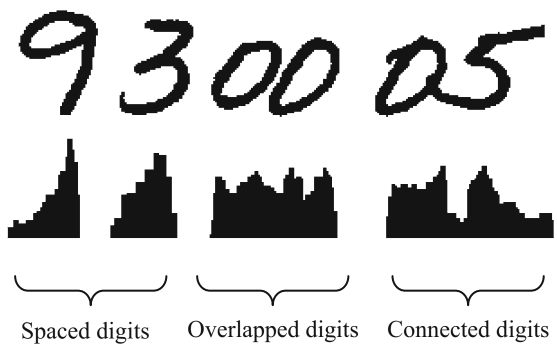


Fig. 1 Correct and incorrect segmentation using the HVP

string with the unknown length. However, when digits are overlapped or connected, the segmentation is not possible. Indeed, the HVP can split the digit string comprising either a single digit or more than two digits. Hence, each isolated digit is well recognized by the classifier. In contrast, when digit components are connected or overlapped, the classifier cannot recognize the digit. In this case, the classifier plays an important role for detecting the overlapped or/and connected digits. Figure 1 illustrates an example of the correct and incorrect segmentation of the digit string when using the HVP.

In some cases, segmented digits are broken in multiple parts (Fig. 2). In this case, the recognition is performed by the classifier without any preprocessing method.

2.2 Segmentation of overlapped digits

The segmentation of overlapped digits is performed when adjacent digits are overlapped. It is based on the contour analysis using the contour detection, which is extracted from the binary image using the mathematical morphology [1, 12, 14, 15, 17]. Hence, two adjacent digits are then separated using a distance termed t_d, which is fixed experimentally. Figure 3 illustrates the possible and impossible segmentation when analyzing the digit contour.

In some cases, when using the contour detection, it appears broken digits which are necessary to process for improving the performance of the segmentation. In this case, the broken parts of an overlapped component are detected by examining the intersection with the median line (i.e., half height) of each component image [18]. Figure 4 presents two examples of the incorrect segmentation.

Generally, a digit is broken into two parts for which we associate each one a contour, namely (C1) and (C2), respectively. Figure 5 shows different configurations of a broken digit. Hence, three rules are specifically used for their detections:

- **Rule 1** When both C1 and C2 intersect the median line, then the broken parts are considered as two distinct overlapped single digits. In this case, these adjacent digits are straightforward separated using a fixed distance (Fig. 5a).
- **Rule 2** When both C1 and C2 do not intersect the median line, then the broken parts are considered belonging to the same single digit (Fig. 5b).
- **Rule 3** When C2 does not intersect the median line and C1 intersects the median line, then C1 and C2 belong to the current digit (Fig. 5c); otherwise, C1 and C2 belong to the next digit (Fig. 5d). For both cases, the belonging of C2 to the current digit or to the next digit is decided by the classifier.

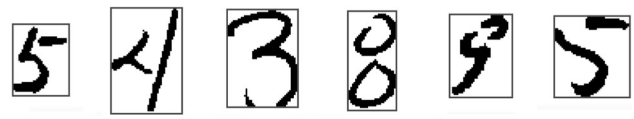


Fig. 2 Sample images of broken handwritten digits

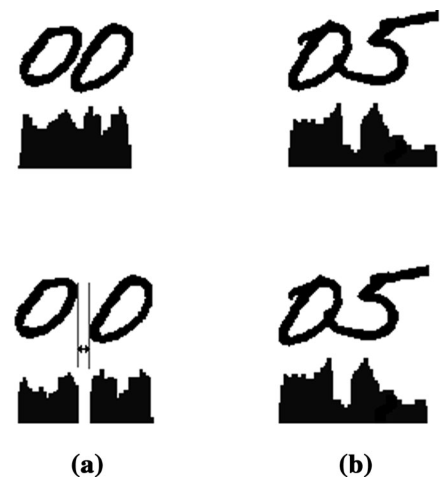


Fig. 3 Segmentation by contour analysis. **a** Possible segmentation. **b** Impossible segmentation

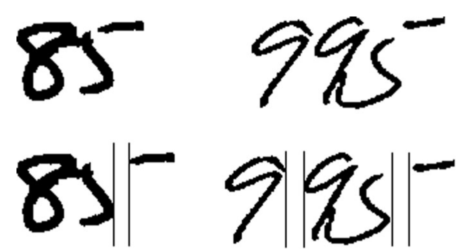


Fig. 4 Examples of incorrect segmentation when using the contour analysis

2.3 Segmentation of connected digits

A great effort has been conducted for developing various techniques to the detection of the connected digits using information on the contour, skeleton and profile of the

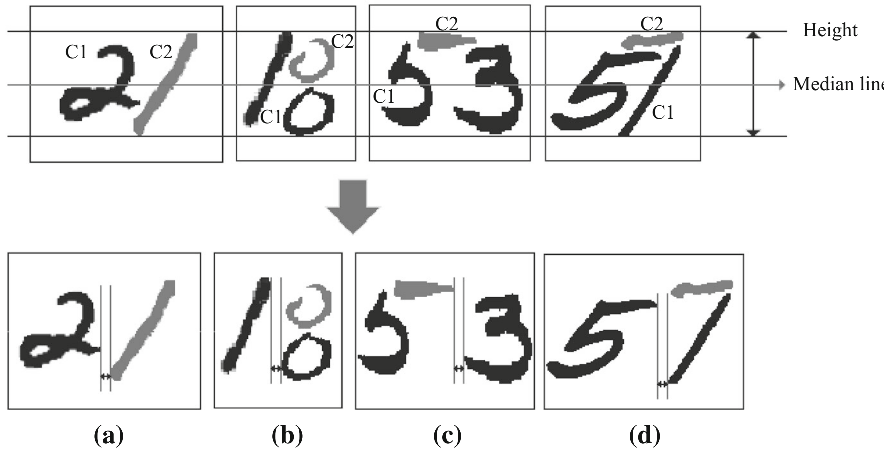


Fig. 5 Processing of broken parts **a** two distinct overlapped single digits (Rule 1), **b** broken single digit (Rule 2), **c**, **d** broken digit (Rule 3)

digit string. For instance, Elnagar and Alhajj [19] proposed to split touching digits using thinning processes. The pattern is thinned to obtain a uniform stroke width, and then, the feature points are extracted from the skeleton and contour as end points, joint points and crossing points. Latter, Suwa [5] proposed an improvement of the method proposed by Elnagar and Alhajj [19] by adopting a thinning-based segmentation strategy for a couple of handwritten connected digit. Kyung et al. [20] proposed using structural features of contours to generate multiple hypotheses. These hypotheses are based on the analysis of the ligature and the characteristics of candidate break points. Lei et al. [21] proposed an analysis of external and internal contours as well as projection analysis. These features are used to define the segmentation lines. Ma et al. [22] described a touching pattern-oriented strategy using drop-fall algorithms [1] according to four possible cutting paths (descend left, descend right, ascend left and ascend right). Wang et al. [23] proposed to generate all possible candidate segmentations of an input string through contour and profile analysis. Recently, Gattal and Chibani [24] proposed a method for finding the base points (BPs) and interconnection points (IPs) on the contour and the skeleton of the connected digits according to the connection configuration. After that, a crossing oriented window is set around IP for finding correctly the cutting path.

In this paper, the algorithm is slightly adapted such that it can segment connected digits with unknown lengths. Also, the Radon transform is used in order to perform on the sliding window for selecting the optimal orientation angle and consequently reducing the number of segmentation cuts. The proposed method uses conjointly the contour and the skeleton of the digit string for detecting point connections, which can be performed according to the following steps:

- **Step 1** Apply a contour detection on the connected digit string in order to detect all possible BPs from the local extrema (minima and maxima).
- **Step 2** Perform the skeleton algorithm in order to detect all possible IPs.
- **Step 3** Find IPs by analyzing the skeleton between connected digits. If IP is detected, a sliding window having the same height as the digit image and a fixed width (W_{SWRT}) is set on IP in the middle of the width (W_{SWRT}). Then, the Radon transform is performed on the sliding window around IP for finding the orientation angle (θ_{cut}) in order to get the best cutting. This window is called the sliding window Radon transform (SWRT). The optimal orientation angle (θ_{cut}) and the width (W_{SWRT}) match a right angle inter-digit into the most cases. Also, it allows reducing the number of segmentation cuts.
- **Step 4** For each SWRT detected in the digit string, a list of segmentation hypotheses is provided in order to find the best cutting path. Therefore, three hypotheses can be considered: If a single IP is found, then the cut is making between single IP with upper BP and lower BP. When two IPs (upper IP and lower IP) are found, then the cut is making between BPs and IPs. In order to avoid under-segmentation, the cut is making between closest points (upper BP and lower BP) in the middle. More details are reported in [24].
- **Step 5** In some cases, digits may be connected in such a way there is no IP on the skeleton leading to failure of positioning the SWRT. In this case, to avoid under-segmentation due to lack of IPs, the cut is made between closest upper and lower BPs in the middle of two connected digits.

The evaluation of each hypothesis is performed using the digit recognition–verification module, which is presented in Sect. 3. Furthermore, the number of the segmented components depends on the number of detected IPs. Hence, two important parameters influence the quality of the segmentation, which are the adjustment of the width (W_{SWRT}) and finding the orientation angle (θ_{cut}) using SWRT.

2.3.1 Adjustment of the width (W_{SWRT})

Adjusting the appropriate width (W_{SWRT}) of the SWRT directly influences the performance of the segmentation. For instance, a large W_{SWRT} generates small number of cutting paths leading to an over-segmentation. In contrast, a small W_{SWRT} generates many cutting paths leading to an under-segmentation. Consequently, a trade-off between small and large width should be defined in order to lose or ignore some important cutting paths. Figure 6 illustrates the influence of adjusting the width of the sliding window.

In our case, the width W_{SWRT} is set as a fraction of its height, i.e., $W_{\mathrm{SWRT}} = \mathrm{Height}/\beta$ where Height is the height of the connected digits and β is a parameter defined experimentally. When the height of connected digits is changed, the width W_{SWRT} is also changed in order to keep the aspect ratio and allows avoiding the problem of variations in the shapes and styles.

2.3.2 Finding the angular cut via the Radon transform

The skew or orientation angle of the connected digits is estimated using the Radon transform of the image [25, 26]. It is computed by projecting the image along specified directions, which allows generating a set of line integrals or slices from multiple sources. More precisely, the Radon transform takes multiple and parallel projections of the image from different angles by rotating the source around the center of the image. Formally, the Radon transform is performed on a binary image, which is defined as [25]:

$$g(r,\theta) = \iint I(x,y) \cdot \delta(x\cos\theta + y\sin\theta - r)\,\mathrm{d}x\mathrm{d}y \qquad (1)$$

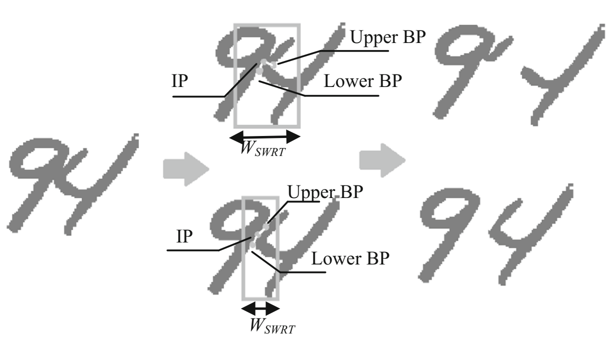


Fig. 6 Influence of adjusting the width on the sliding window

where δ is the Dirac function, $\theta \in]0, 179°]$ and $r \in]-\infty, +\infty]$. In other words, $g(r,\theta)$ is the integral of $I(x,y)$ over the line defined by $\rho = x\cos\theta + y\sin\theta$.

Figure 7 shows some illustrative examples of the Radon transform computed on the binary connected digit images. The high brightness indicates the maximum amplitude of the Radon transform. Therefore, the orientation angle, which defines the angular cut (θ_{cut}), is deduced by finding the maximum value of the Radon transform $g(r,\theta)$. Figure 8 shows the Radon slices for two selected projection angles. The angular cut (θ_{cut}) is selected according to the amplitude of the Radon slice. The slope of the SWRT fixed around IP allows getting a better cutting path. Figure 9 shows the impact of selecting the orientation angle around IPs for segmentation with and without using the SWRT.

3 Design of the digit string recognition system

The proposed digit string recognition system allows examining each type of the segmentation problem (spaced, overlapped or connected digits) in order to select the appropriate segmentation method. As shown in Fig. 10, its design requires three main modules, which are the primary segmentation, secondary segmentation and digit recognition–verification. The digit string image is primarily segmented using HVP, which produces multiple segmented components (SC); each one is verified according to its size using the segmented component analysis (SCA). If the size is greater than a fixed threshold, SC is considered as a digit (isolated digit). Therefore, it is submitted to digit recognition–verification (DRV) module for accepting or rejecting the digit. In contrast, if SC is rejected by DRV, the secondary segmentation is performed to separate the overlapped and/or connected digits. In this case, the contour detection is performed on SC considered as a nondigit, which is analyzed and verified using successively contour analysis (CA), SCA and DRV, respectively. Otherwise, SWRT technique and grouped component analysis (GCA) are used conjointly for segmenting SC. GCA uses DRV for evaluating each grouped component (GC) according to the configuration link in order to provide optimal decisions in difficult situations. In the following, we first describe the design of DRV and then the steps for segmenting the digit string image.

3.1 Digit recognition–verification

The digit recognition–verification module is composed of two submodules, which are the digit recognition based on feature generation module and the classification module, and the digit verification. Their design directly affects the robustness of the proposed handwritten digit string recognition. In the following, we briefly describe feature

Fig. 7 Radon transform for three examples of connected digits. The maximum value of the Radon transform corresponds to the orientation angle θ_{cut} for cutting two connected digits

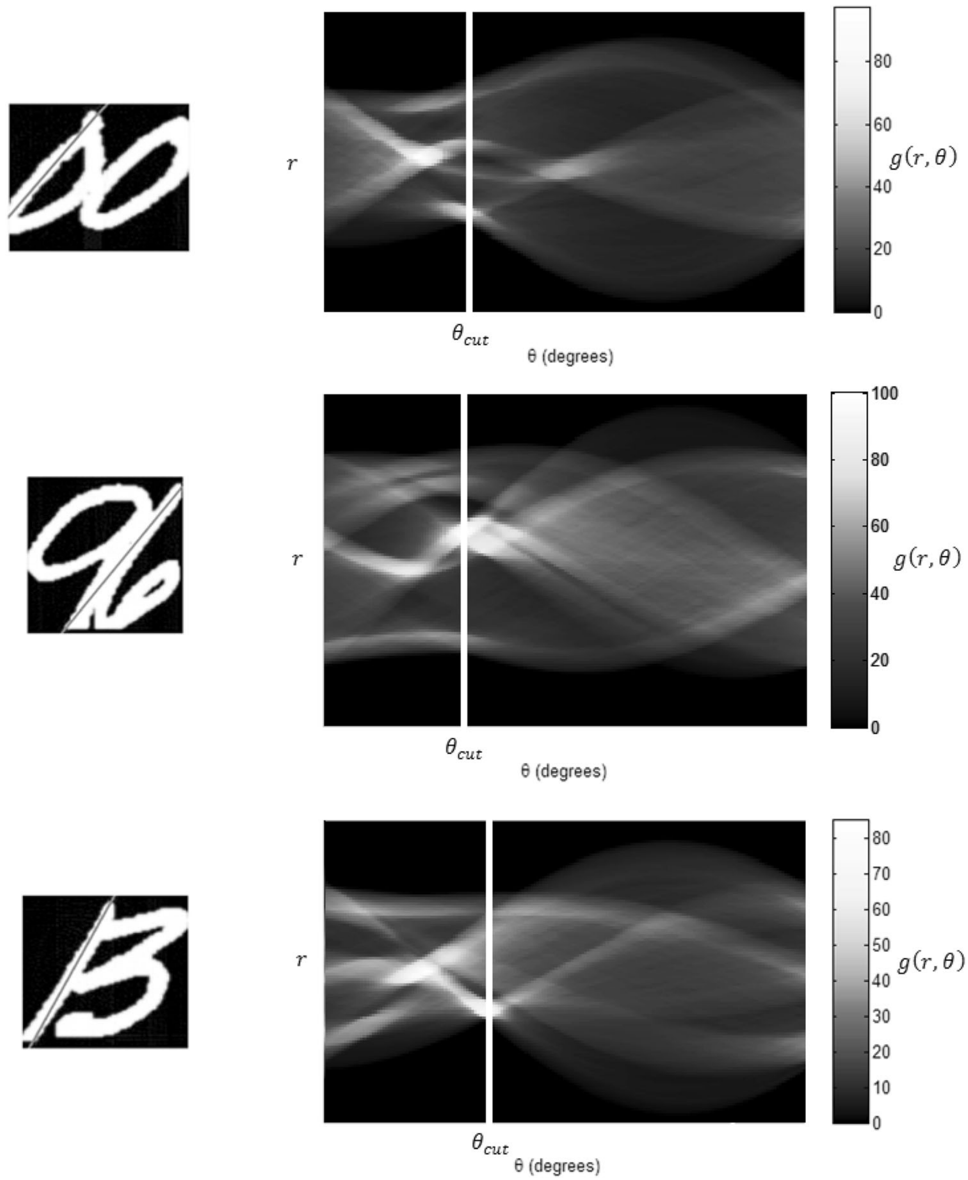
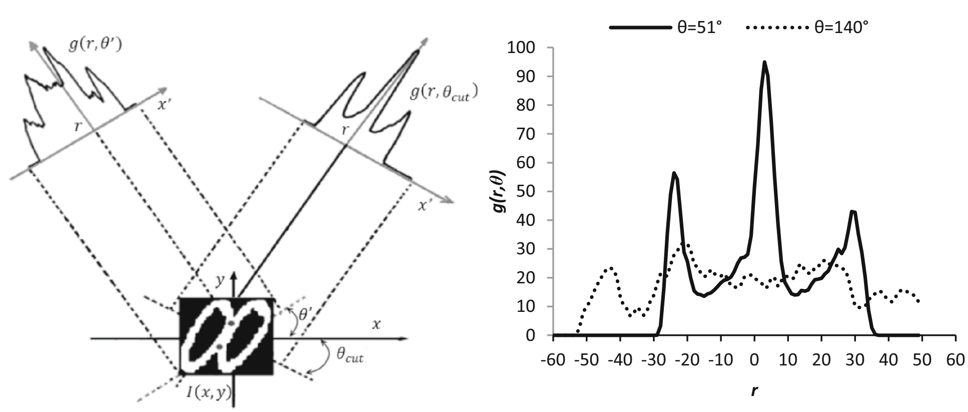


Fig. 8 Two selected projections of the Radon transform showing $\theta_{\mathrm{cut}} = 51^{\circ}$



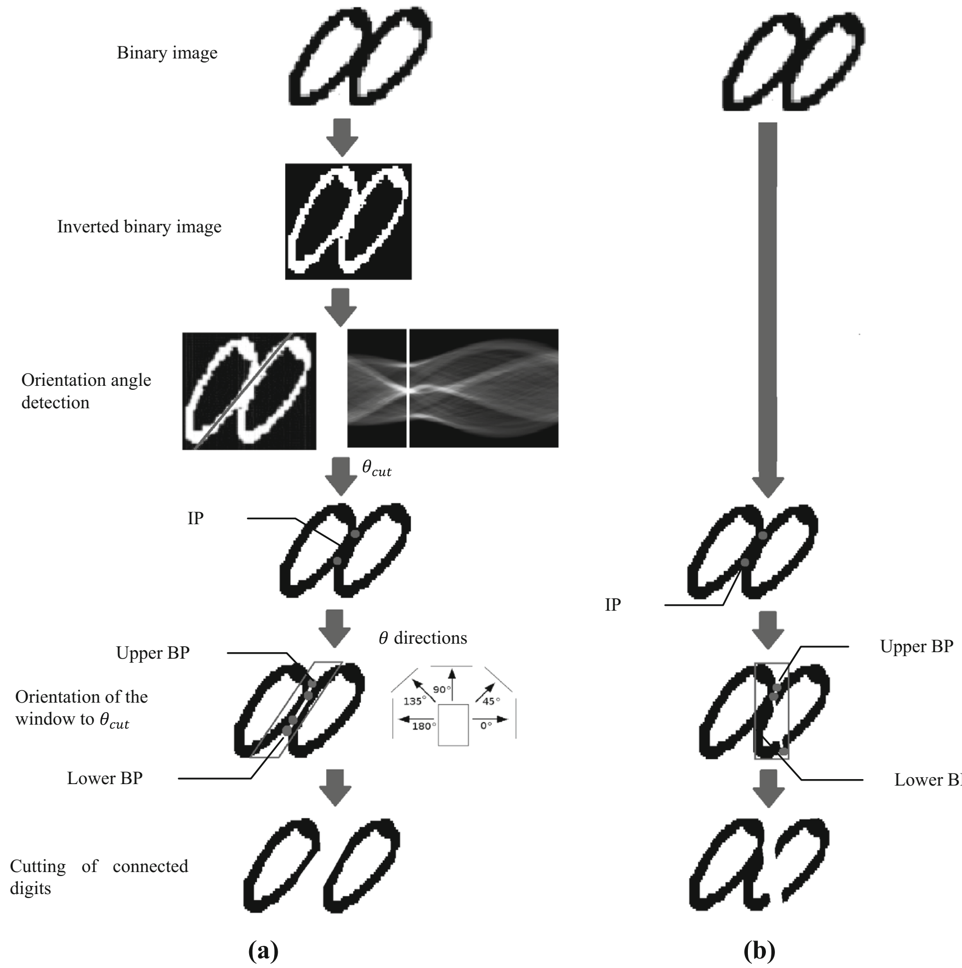


Fig. 9 Impact of selecting the orientation angle for segmentation. **a** Segmentation with SWRT. **b** Segmentation without SWRT

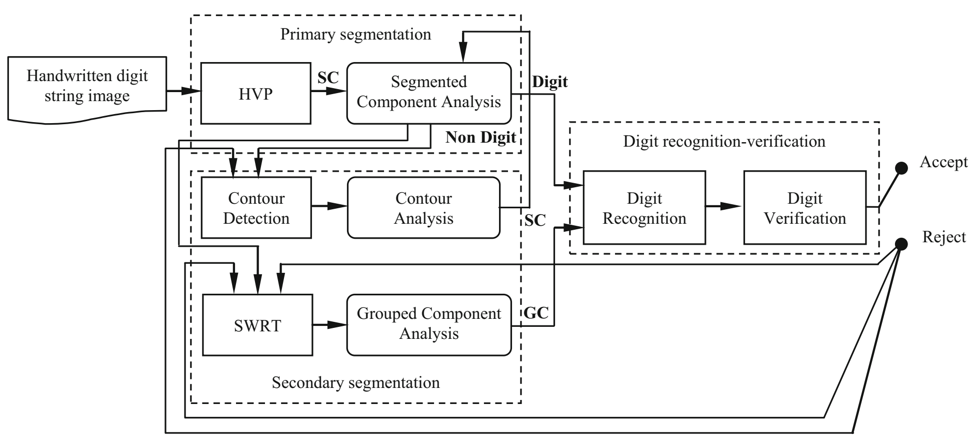


Fig. 10 Full segmentation system for handwritten digit string recognition

generation method, classifiers and the technique of verifying the digit.

3.1.1 Feature generation

Various feature generation methods have been proposed in recent years. In the proposed system, we use a combination of multiple features for improving the overall recognition rates by minimizing the intra-class variability and maximizing inter-class variability. These features used in the work [27] include some global statistics, moments, profile and projection-based features and features computed from the contour and skeleton of the digit. Some of these features are extracted from the whole image of the digit, while others are extracted from different regions (4 regions) of the image by first applying a uniform grid sampling to the image. Hence, we notice that the uniform grid sampling and used features are well suited for discriminating isolated digits from under-segmented ones.

3.1.2 Design of the SVM classifiers on isolated digits

The classification module is based on the well-known binary support vector machine (B-SVM) classifiers implemented through the one-against-all (OAA) architecture. Ten (10) B-SVMs are then used for recognizing an isolated digit from the set $\Omega = \{\omega_0, \ldots, \omega_9\}$ such that ω_j is the class label of a digit $j = 0, \ldots, 9$. Each B-SVM classifier is trained using the radial basis function kernel for separating a digit class from other classes.

More precisely, each digit is associated with a SVM classifier providing a response termed $f_j(x_{SC}), j = 0, \ldots 9$ such that x_{SC} is the feature vector of the segmented component (SC). Therefore, SC is assigned to one of the classes Ω according to the following decision rule:

$$f_{\max}(x_{SC}) = \max\{f_j(x_{SC}); j = 0, \ldots, 9\} \tag{2}$$

$f_{\max}(x_{SC})$ is the maximal value selected from 10 responses provided by SVM classifiers.

3.1.3 Digit verification

The maximal value selected from 10 SVM responses does not ensure that SC is a true isolated digit. Indeed, in some cases, SC submitted to the recognition can be a connected or overlapped digit. In order to ensure a better recognition of the digit, we propose to add a threshold termed t_f as a verifier using the following decision rule: x_{SC} is accepted as a digit if $f_{\max}(x_{SC}) \geq t_f$; otherwise, it is rejected. Conversely, when x_{SC} is rejected (i.e., $f_{\max}(x_{SC}) < t_f$), SC is then considered as a nondigit (overlapped and/or connected digit).

The threshold t_f is deduced during training of the SVMs by taking the minimum of the SVM maximum outputs from all isolated digits, let:

$$t_f = \min\{\max\{f_j(x_i); j = 0, \ldots 9\}; i = 1, \ldots N\} \tag{3}$$

x_i is the feature vector of the isolated digit and N is the number of isolated digits used for training the 10 SVM classifiers.

3.2 Spaced digit recognition–verification

The spaced digit recognition–verification is performed using successively the HVP, SCA and DRV. The HVP allows producing multiple SCs; each one is analyzed by SCA in order to decide whether it is a digit or nondigit. In our case, SC is considered as an isolated digit by exploiting the ratio between its height and width using the following decision rule:

$$x_{SC} \in \begin{cases} \text{Digit} & \text{if } R_{HW} \geq t_{HW} \\ \text{Non Digit} & \text{otherwise} \end{cases} \tag{4}$$

Such that $R_{HW} = \text{Height}/\text{Width}$ and t_{HW} is a fixed decision threshold. Consequently, SC is considered as an isolated digit if $R_{HW} \geq t_{HW}$; otherwise, it is considered as an overlapped or connected digit. When SC is detected as an isolated digit, it is submitted to the DRV for its recognition; otherwise, it is submitted to the secondary segmentation module. The ratio of SC is considered as additional information that helps the classifier to avoid the under-segmentation problem.

In some cases, connected digits are verified as isolated digits by SCA since their ratio is greater than a threshold (i.e., $R_{HW} \geq t_{HW}$) as shown for instance in Fig. 11. To avoid misrecognition, SC is accepted definitively using the decision rule:

$$x_{SC} \in \begin{cases} \text{Accept} & \text{if } R_{HW} \geq t_{HW} \text{ and } \quad f_{\max}(x_{SC}) \geq t_f \\ \text{Reject} & \text{otherwise} \end{cases} \tag{5}$$

When SC is rejected, it is submitted to the secondary segmentation module.

3.3 Overlapped digit recognition–verification

The overlapped digit recognition–verification is performed when the ratio R_{HW} of SC is less than a threshold (t_{HW}). In this case, SC is considered as nondigit and can contain more than one overlapped digits. Therefore, a contour detection is performed on SC for their separation using a fixed threshold (t_d) and some specific rules (see 2.2). The resulting subcomponents are verified by SCA for deciding whether each one is a digit or nondigit. When it is detected as a digit, DRV is used for accepting or rejecting it. In the case of rejecting, the segmentation method based on SWRT and GCA is performed to separate two or more connected digits according to the connection type.

3.4 Connected digit recognition–verification

The connected digit recognition–verification uses conjointly SWRT method, GCA and DRV. When using SWRT, the connected digits are split into a sequence of

Fig. 11 Some samples of connected digits considered by SCA as isolated digits

segments; each one is considered as a segmentation hypothesis, which is expected to contain a digit or a fragment of a digit. In this case, all segmentation hypotheses can be represented through a segmentation graph [13, 28]. More specifically, the recognition–verification of connected digits can be performed into two steps:

- Generate all possible segmentation hypotheses on the connected digit string using SWRT. Each hypothesis defines an isolated digit or a fragment of a digit, which is verified by DRV.
- Perform GCA in order to enable the duplication of segments.

Hence, GCA allows grouping the sequence of segments for generating grouped components (GC), which are submitted to DRV for accepting or rejecting. If GC is accepted, then it is considered as a digit; otherwise, it is considered as a nondigit. The grouping of segments using conjointly GCA and DRV is performed according to the following heuristic rules:

- GC is submitted to DRV only if GC intersects the median line.
- If GC intersects the median line and not accepted by DRV, then it is grouped with the previous or next segment and tested with DRV using the following decision rule:

$$x_{GC} \in \begin{cases} \text{Accept} & \text{if GC intersects median line and } f_{\max}(x_{\text{GC}}) \geq t_f \\ \text{Reject} & \text{otherwise} \end{cases} \tag{6}$$

  The reject of GC is considered as an unlikely segmented component. In this case, it is removed from the list of hypotheses.
- When GC does not intersect the median line, then it is grouped with the previous or next segment component and tested with DRV, and so on.

Figure 12 shows an example of using the heuristic rules for grouping the segments, while Fig. 13 depicts an illustrative example of segmenting a digit string image.

4 Experimental results

In order to show the effect of using multiple segmentation methods on unknown length of the digit string, the well-known NIST SD19 database [29] is used containing isolated digits and digit strings. Experimental results are conducted for evaluating the primary and secondary segmentations as well as the overall segmentation. Since the segmentation of the connected digits affects considerably the performance of the digit string recognition, a careful evaluation is conducted to study the influence of the sliding window Radon transform parameters. All best obtained results are then compared to the state of the art.

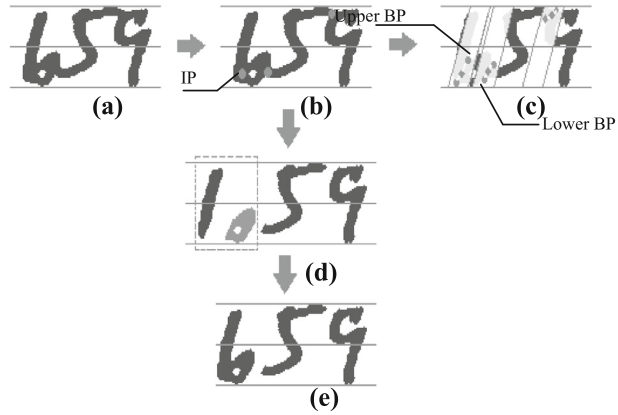


Fig. 12 Connected digit recognition–verification. **a** Original connected digits. **b** Scan IPs. **c** Fixing sliding window Radon transform around IP. **d** Segmentation paths. **e** Final decision

4.1 Databases and evaluation criteria

For a fairly evaluation and comparison with the existing similar algorithms, we use the same database NIST SD19 as used in Refs. [6, 14–16]. Hence, two digit sample sets are extracted from NIST SD19. The first one containing 90,000 handwritten isolated digits is used for building DRV since it has more variations in their shapes and styles, while the second one containing 12,802 digit string images is extracted from the hsf_7 series NIST NSTRING SD19 database, which is distributed into six categories according to their lengths: 2-digit (2370), 3-digit (2385), 4-digit (2345), 5-digit (2316), 6-digit (2169) and 10-digit (1217) strings, respectively. These data exhibit different difficulties such as sloping, overlapping and connecting as well as different string lengths. Table 1 shows the distribution of the database according to the various difficulties.

The evaluation criterion used for conducting experiments and comparison is based on the recognition rate (Rec. Rate) performed on the six digit string lengths: 2-digit, 3-digit, 4-digit, 5-digit, 6-digit and 10-digit strings, respectively.

All recognized segmented components provided by the proposed system are compared to the ground truth. If the segmented component is assigned to the corresponding ground truth, then the segmentation is considered as successful.

The evaluation of the proposed system is conducted by providing in each time the recognition rate obtained for

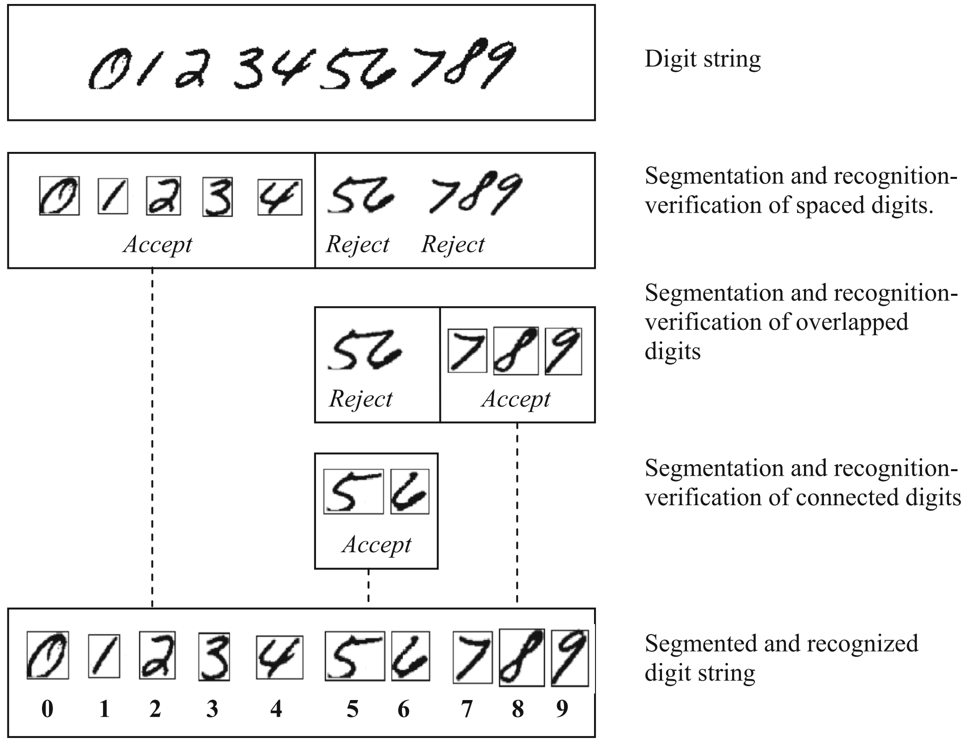


Fig. 13 Segmentation example of our proposed digit string recognition system

Table 1 Number of digit string samples (#Strings) distributed according to the numbers of spaced digits (#SD) and connected and/or overlapped digits (#C–OD) expressed also in %

String length	#Strings	#SD	#C–OD	SD (%)	C–OD (%)
2-digit	2370	1656	714	69.87	30.13
3-digit	2385	1303	1082	54.63	45.37
4-digit	2345	1219	1126	51.98	48.02
5-digit	2316	1125	1191	48.58	51.42
6-digit	2169	1202	967	55.42	44.58
10-digit	1217	679	538	55.79	44.21
Total	12,802	7184	5618	56.12	43.88

primary and secondary segmentations as well as the overall rate for each string length.

4.2 Experimental setup

The performance of the proposed digit string recognition system depends on the robust design of DRV on isolated digits, the appropriate selection of the decision threshold (t_f) used for accepting or rejecting the digit, the threshold t_{HW} used for deciding whether SC is a digit or nondigit, the distance (t_{d}) between two overlapped digits, β used for adjusting the width of the sliding window (W_{SWRT}) and the adjustment of the Radon transform parameters (range of the projection angle and the angular step). t_f and t_{HW} are deduced from isolated digits, whereas t_{d} and β are fixed from the digit string images. Therefore, all these parameters are fixed as follows:

- $t_f = 0.2$: t_f is selected by taking the minimum value of the maximum SVM outputs among recognized 90,000 isolated digits.
- $t_{\mathrm{HW}} = 2$: t_{HW} is deduced from 90,000 isolated digits by taking the maximal ratio of R_{HW}.
- $t_{\mathrm{d}} = 5$ pixels: t_{d} is fixed by examining the overlapped digits contained in NSTRING SD19.
- $\beta = 5$: W_{SWRT} is selected as the height fifth of two connected digits by examining all connected digits contained in NSTRING SD19.

4.3 Design of SVM classifiers on isolated digits

The performance of a digit string recognition system depends in particular on the design of a robust SVM classifier on isolated digits. In our case, the recognition module is based on the SVM multi-class approach using the one-against-all implementation [30]. Two important parameters are required for training the SVM classifiers, which are the regularization parameter (C) and the radial basis function (RBF) kernel parameter (σ). Hence, SVM classifiers are trained and their parameters are adjusted by using the NIST SD19 handwritten isolated digit database.

The efficiency of the SVM classifiers also depends on the number of samples used for building a robust DRV. Thus, 90,000 handwritten digits are divided into subsets, which are used for training (60,000 images) and validating (30,000 images) the SVM classifier parameters (C,σ), respectively. When parameters are found, the SVM classifiers are retrained by considering all handwritten isolated digit images (90,000 images). More precisely, training and retraining of the SVM classifiers are performed according to the two following steps:

- **Training** During this step, C and σ are varied in the interval [1, 100] and [1, 50], respectively. The optimal couple is selected when the validation rate is maximal. In our case, the optimal C and σ parameters are set to 18 and 11, respectively.
- **Retraining** After training, SVM classifiers are retrained using all samples (training and validation, i.e., 90,000 samples) by keeping the same parameters ($C = 18$ and $\sigma = 11$).

Table 2 reports the performance of the proposed segmentation and recognition system when performing training and retraining SVM classifiers. The recognition rates (%) are reported for the primary and secondary segmentations as well as the overall recognition for each string length. The projection angle θ of the Radon transform is ranged in [1, 179], while the value of the angular step is set to 10°.

We clearly can notice that the retraining of the SVM classifiers improves the best overall recognition rate against the training of the SVM classifiers. Indeed, the overall recognition rate when retraining the SVM classifiers significantly increases whatever the string length. This improvement is due to the robustness of SVM classifiers for building an efficient DRV. Furthermore, we can observe that the best improvement is specifically offered for the secondary segmentation compared to the primary segmentation. This means that the recognition of spaced digits is more simple than the recognition of overlapped or/and connected digits.

4.4 Evaluation of the digit segmentation

The performance of the segmentation is strongly influenced by the presence of connected digits into the digit string. In the proposed system, the segmentation of connected digits is performed using the Radon transform and its appropriate parameters for finding the best cutting path. The two

Table 2 Recognition rate (%) obtained when training and retraining SVM classifiers

String length	#Strings	Rec. rate when training SVMs (%)			Rec. rate when retraining SVMs (%)		
		Primary	Secondary	Overall	Primary	Secondary	Overall
2-digit	2370	99.73	79.55	93.65	99.88	80.46	94.03
3-digit	2385	99.49	85.67	93.22	99.87	86.94	94.00
4-digit	2345	98.69	86.92	93.04	99.20	88.45	94.04
5-digit	2316	97.69	88.38	92.90	98.31	89.76	93.91
6-digit	2169	98.13	87.81	93.53	98.68	90.06	94.84
10-digit	1217	97.81	86.36	92.74	98.38	88.64	94.08
Overall	–	98.59	85.78	93.18	99.05	87.38	94.15

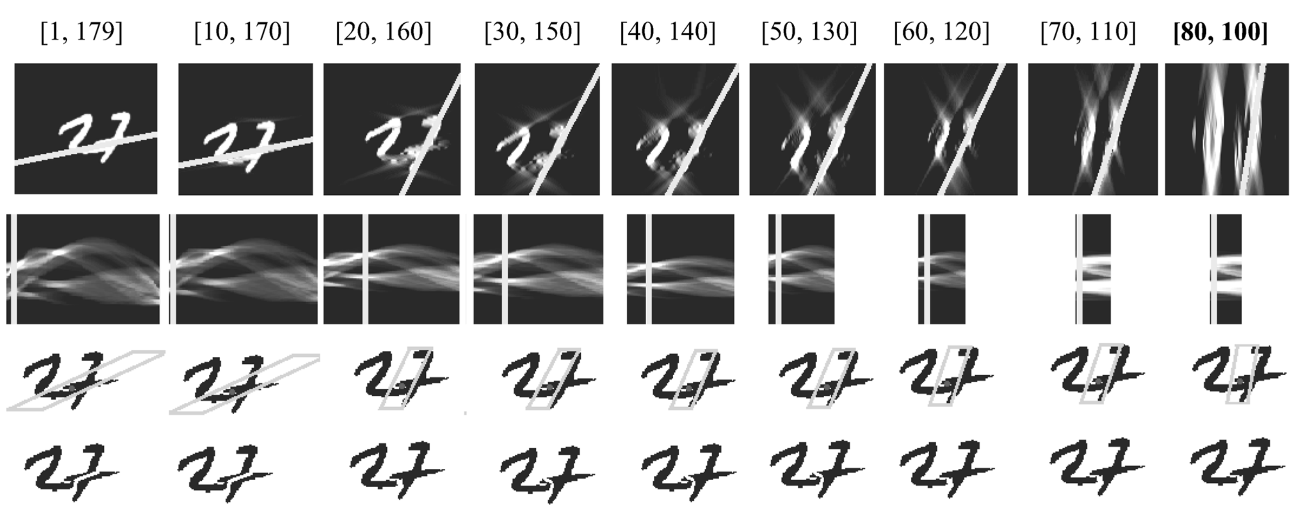


Fig. 14 Impact of selecting the range of the projection angle for detecting the cutting path from [1, 179] to [80, 100]

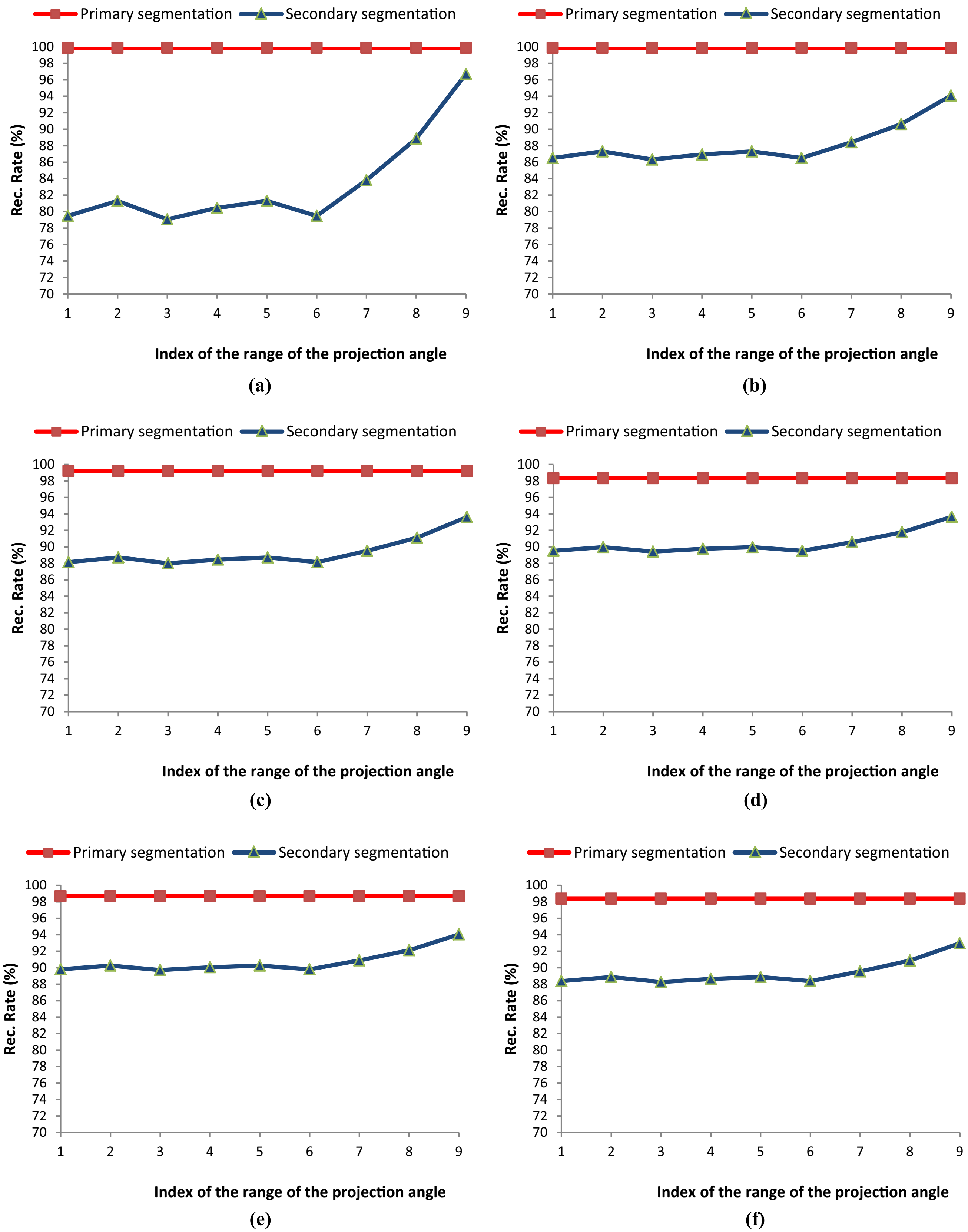


Fig. 15 Influence of selecting the range of the projection angle (°) for different digit string lengths: **a** 2-digit, **b** 3-digit, **c** 4-digit, **d** 5-digit, **e** 6-digit and **f** 10-digit

parameters are, respectively, the range of the projection angle (θ) and the value of the angular step. In the following, these two parameters are studied for evaluating their impact of the digit string recognition.

4.4.1 Influence of the range of the projection angle

Selecting the range of the projection angle plays an important role for finding the best cutting path. Figure 14 depicts an illustrative example showing the influence of selecting the range of the projection angle when segmenting two connected digits using SWRT. As we can see, when the range of the projection angle is reduced, the cut is well performed.

The recognition rates for the primary and secondary segmentations are reported in Fig. 15 according to the range of the projection angle when the value of the angular step is fixed to 10°. Table 3 reports the average rates for all digit string lengths derived from different ranges of the projection angle. The last column reports the overall recognition rate for all string lengths provided by the proposed system with respect to the range of the projection angle (°), while the value of angular step is set to 10°.

As we can see, the selection of the projection angle affects the performance of the system more specifically for the secondary segmentation whatever the length of the digit string. Moreover, the recognition rates for each digit string length and the average rates for all digit string lengths are improved when the range of the projection angle is reduced.

The best performance is obtained when the range of the projection angle is fixed in the interval [80, 100]. For the primary segmentation, the selected range does not affect the performance of the system since SWRT is performed only on the secondary segmentation. In contrast, this selected range allows improving the recognition rate till 94.18% for the secondary segmentation providing an overall recognition rate of 96.87%. The good performance

Table 3 Average rates obtained for all digit string lengths derived from different ranges of the projection angle (°)

Range of the projection angle (°)	Rec. rate (%)		
	Primary	Secondary	Overall
[1,179]	99.05	86.98	93.99
[10,170]	99.05	87.74	94.29
[20,160]	99.05	86.80	93.92
[30,150]	99.05	87.38	94.15
[40,140]	99.05	87.74	94.29
[50,130]	99.05	86.98	93.99
[60,120]	99.05	88.79	94.71
[70,110]	99.05	90.90	95.56
[80,100]	99.05	94.18	96.87

obtained for this range of angles is fairly predictable since the writing generated by a writer is generally inclined around at an angle about 90°. Hence, this range of angles for SWRT can be considered as an optimum for all digit strings.

4.4.2 Adjustment of the angular step

The adjustment of the angular step for SWRT is also considered as a crucial parameter for achieving a robust system. Its value is a trade-off between the computation time and the recognition rate. In order to study its influence, the proposed system is evaluated for different values of the angular step ranging from 1° to 10°. The range of the projection angle is fixed in [80,100] according to the best performance obtained previously. Figure 16 depicts the recognition rate versus the angular step for different digit string lengths. Table 4 reports the average rates for each angular step computed for all digit string lengths according to the primary and secondary segmentations, respectively. The last column reports the overall recognition rate for all string lengths provided by the proposed system with respect to the angular step.

As we can see, adjusting the angular step from 1 to 10 (°) does not affect considerably the performance of the proposed system whatever the length of the digit string (Fig. 16). However, the best improvement is obtained when the angular step is fixed to 2 or 3 providing an average recognition rate 94.29% for the secondary segmentation and an overall recognition rate of 96.91% (Table 4).

In order to show the impact of selecting the appropriate angular step, Fig. 17 illustrates the correct and incorrect segmentation when the angular step is fixed to 3° and 6°, respectively. When the angular step is correctly adjusted (Fig. 17a), the upper BP and lower BP are well detected into the SWRT and therefore the separation of the two adjacent digits is well performed. In contrast, when the angular step is incorrectly adjusted (Fig. 17b), the upper BP and lower BP are not well detected, which leads to an incorrect segmentation since the SWRT is fixed around IP on the wrong orientation angle.

4.5 Comparative analysis

In order to compare the proposed system against others using the same digit string length image database (NIST NSTRING SD19) [6, 14–16], four systems are selected according to the state of the art, which are proposed by Oliveira et al. [14], Britto et al. [6], Oliveira and Sabourin [15] and Sadri et al. [16]. Table 5 reports the recognition rates obtained for different systems including the proposed

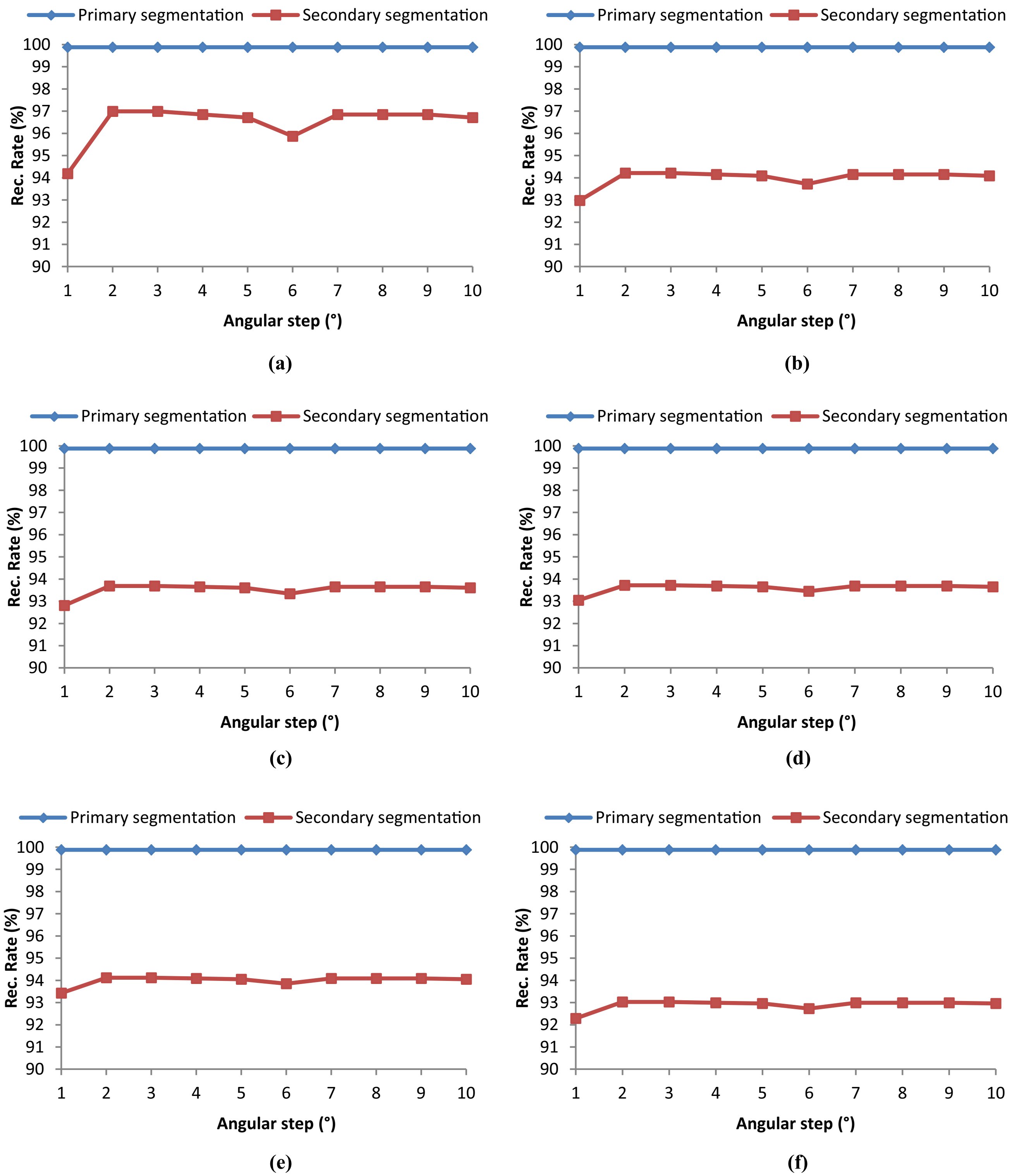


Fig. 16 Influence of adjusting the angular step (°) for different digit string lengths. **a** 2-digit string length, **b** 3-digit string length, **c** 4-digit string length, **d** 5-digit string length, **e** 6-digit string length and **f** 10-digit string length

one for an unknown-length string performed on NIST NSTRING SD19. As we can see, the proposed system achieves an overall average rate of 96.91% for all string lengths and outperforms the state-of-the-art results more specifically when the string length is large (6- and 10-digits).

Table 4 Recognition rates according to the angular step for each digit string length

Angular step (°)	Rec. rate (%)		
	Primary	Secondary	Overall
1	99.05	93.12	96.45
2	99.05	**94.29**	**96.91**
3	99.05	**94.29**	**96.91**
4	99.05	94.24	96.89
5	99.05	94.18	96.87
6	99.05	93.83	96.73
7	99.05	94.24	96.89
8	99.05	94.24	96.89
9	99.05	94.24	96.89
10	99.05	94.18	96.87

Bold values indicate significant recognition rate

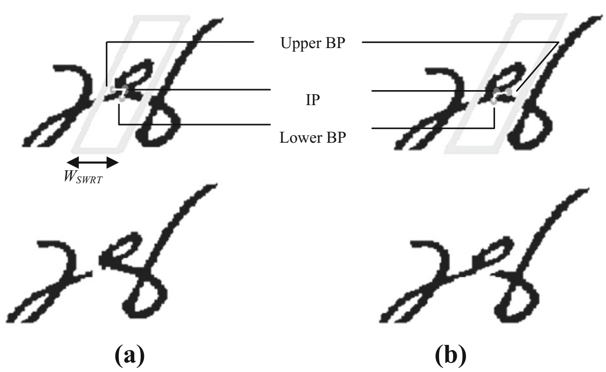


Fig. 17 Impact of selecting the angular step for detecting the cutting path: **a** correct segmentation when the angular step is fixed to 3°. **b** Incorrect segmentation when the angular step is fixed to 6°

Table 6 depicts some examples of the correct and incorrect recognition of the digit strings provided by the proposed system. As showed, three possible situations occurred when performing the segmentation–recognition: correct segmentation and correct recognition, incorrect segmentation and/or incorrect recognition. The incorrect segmentation is due to the difficulty of SWRT to find IP for cutting to adjacent digits. In contrast, the incorrect recognition is due to the wrong writing of the digit.

4.6 Computation cost of the proposed system

The proposed digit string recognition system allows resolving different segmentation problems in order to select the most appropriate one. Since this system can be deployed in real environment, it is interesting to compute the cost required for treating all situations. In this way, the computational cost provides good insight into the complexity of the system which can be expressed by the number of segmentation hypotheses produced by the system, the number of heuristics and complexity of the features used to yield the segmentation cuts. However, using the classifier for analyzing and deciding the rejection or acceptance each segmented digit image leads a very high computational cost.

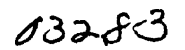
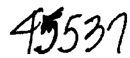
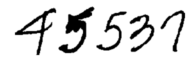
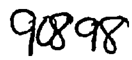
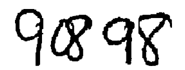
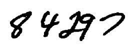
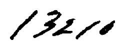
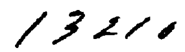
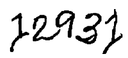
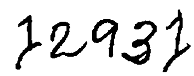
In order to compute the computational time of the existing similar systems, an objective evaluation would be performed using the same coding standard and the same hardware. Therefore, it is impossible to compute this metric without codes for these systems. Hence, the performance comparison of the algorithms in terms of the computational cost is more appropriate by using information provided in their respective publications.

In this context, Britto et al. [6] did not take into account the computation time during the experiments. However, the N-best algorithm has been evaluated as the most time module of the proposed method. Besides, the computational cost in Oliveira et al. [14] computed incrementally the k-best path with the shared subpaths going through each node of the graph. Afterward, each hypothesis (best path) is submitted to the post-processor module, which checks whether it satisfies the application rules or not. Indeed, in terms of running time, Oliveira and Sabourin [15] showed that SVMs have a higher computational cost than MLP neural networks [14] in isolated digit recognition. Regarding computational cost in Sadri et al. [16], there are 2^n possible

Table 5 Comparative analysis of various segmentation systems for an unknown-length string performed on NIST NSTRING SD19

String length	#Strings	Oliveira et al. [14]	Britto et al. [6]	Oliveira and Sabourin [15]	Sadri et al. [16]	Our approach
2-digit	2370	96.88	94.80	97.67	95.05	99.01
3-digit	2385	95.38	91.60	96.26	91.43	97.30
4-digit	2345	93.38	91.30	94.28	91.07	96.56
5-digit	2316	92.40	88.30	94.00	88.05	95.95
6-digit	2169	93.12	89.10	93.80	88.69	96.65
10-digit	1217	90.24	86.90	91.38	86.13	96.01
Overall	–	93.57	90.33	94.57	90.07	96.91

Table 6 Examples of the correct and incorrect segmentation–recognition produced by the proposed system from the NSTRING SD19 database

	Digit string	Class labels	Segmented digits	Assigned classes
Correct segmentation and recognition		03283		03283
		75,434		75,434
		45,537		45,537
		5601		5601
Incorrect segmentation		9898		90,898
		84,097		84,297
Incorrect recognition		13,211		13,210
		72,937		12,931

segmentation hypotheses corresponding to candidate lattice graph from the starting node to the ending node for n cutting paths in a string image. Each of those paths is represented by binary chromosome containing n genes.

In our case, the computation cost is measured by considering three situations:

- **Spaced digit** The total number of segmentation hypotheses using HVP (N_{HVP}) is the same as the number of the spaced digits.
- **Overlapped digit** Let N_{GBC} be the number of grouped broken component provided by CA when the broken component exists and N_{ID} the total number of isolated digit provided by CA; then, the total number of segmentation hypotheses from CA (N_{CA}) is $N_{CA} = N_{GBC} + N_{ID}$
- **Connected digit** Let N_{GC} be the number of GC and N_{SC} the total number of SC; then, the total number of segmentation hypotheses from SWRT (N_{SWRT}) is calculated as $N_{SWRT} = N_{GC} + N_{SC}$.

Finally, the total number of segmentation hypotheses (N_{SH}) is the sum of all numbers of segmentation hypotheses taking the following form:

$$N_{SH} = N_{DS}(N_{HVP} + N_{CA} + N_{SWRT}) \tag{7}$$

where N_{DS} is the number of digits contained in a string. By substituting N_{HVP}, N_{CA} and N_{SWRT}, Eq. (7) becomes:

$$N_{SH} = N_{DS} \cdot (N_{HVP} + N_{GBC} + N_{ID} + N_{GC} + N_{SC}) \tag{8}$$

Equation 8 clearly indicates that the computation cost depends on the digit string length and the complexity of the connected digits for finding the best cutting path.

Besides, the proposed system makes mainly the extensive use of the heuristic rules and SVM classifiers, which contribute to the high computational cost.

5 Conclusion

The goal of this paper was to present a full explicit segmentation system for handwritten digit strings based on a combination of several segmentation methods depending on the configuration link between digits. Hence, three segmentation methods have been combined, which are HVP, CA and SWRT used conjointly with SCA and DRV for verification and recognition.

The benefit of the proposed system is its ability to properly segment the spaced, sloped, overlapped and connected digits without any information about the length of the string when performing successively CA, SCA and DRV for analyzing and verifying the SC.

The obtained performance of the proposed segmentation–recognition system of digit strings depends on three different factors: the accuracy of SVM-OAA and robustness of the DRV, the used segmentation methods and, finally, the combination strategy with some heuristic rules.

In some cases, the unknown length of the digit strings makes the task for classifiers more difficult to recognize many outliers leading to treat, for example, digit strings as valid digits [16]. However, by combining HVP, CA and SWRT, the proposed system allows reducing the confusion of the segmentation–recognition.

For future works, it is interesting to explore the way to use a filter for eliminating the unnecessary segmentation hypothesis in order to reduce the computation time.

References

1. Congedo G, Dimauro G, Impedovo S, Pirlo G (1995) Segmentation of Numeric Strings. In: Proceedings of Third international conference on document analysis and recognition (ICDAR-3). Montreal, pp 1038–1041
2. Sadri J, Jalili MJ, Akbari Y, Foroozandeh A (2014) Designing a new standard structure for improving automatic processing of Persian handwritten bank cheques. Pattern Anal Appl 17(4):849–862
3. Kaufmann G, Bunke H (2000) Automated reading of cheque amounts. Pattern Anal Appl 3(2):132–141
4. Lam L, Suen CY (1988) Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. Pattern Recogn 21(1):19–31
5. Suwa M (2005) Segmentation of connected handwritten numerals by graph representation. In: Proceedings of eighth international conference on document analysis and recognition (ICDAR-05), vol 2. Seoul, pp 750–754
6. Britto AS, Sabourine R, Bortolozzi F, Suen C (2003) Recognition of handwritten numeral strings using a two-stage HMM-based method. Int J Doc Anal Recognit 5(2–3):102–117
7. Shridhar M, Badreldin A (1986) A recognition of isolated and simply connected handwritten numerals. Pattern Recogn 19(1):1–12
8. Shi Z, Govindaraju V (1997) Segmentation and recognition of connected handwritten numeral strings. Pattern Recogn 30(9):1501–1504
9. Fujisawa H, Nakano Y, Kurino K (1992) Segmentation methods for character recognition: from segmentation to document structure analysis. Proc IEEE 80(7):1079–1091
10. Alessandro V, Luettin J (2001) A new normalization technique for cursive handwritten words. Pattern Recogn Lett 22(9):1043–1050
11. Lu Z, Chi Z, Siu WC, Shi P (1999) A background-thinning-based approach for separating and recognizing connected handwritten digit strings. Pattern Recogn 32(6):921–933
12. Pal U, Belaid A, Choisy Ch (2003) Touching numeral segmentation using water reservoir concept. Pattern Recogn Lett 24:261–272
13. Chen YK, Wang JF (2000) Segmentation of single or multiple touching handwritten numeral strings using background and foreground analysis. IEEE Trans Pattern Anal Mach Intell 22(11):1304–1317
14. Oliveira LS, Sabourin R, Bortolozzi F, Suen CY (2002) Automatic segmentation of handwritten numerical strings: a recognition and verification strategy. IEEE Trans Pattern Anal Mach Intell 24(11):1438–1454
15. Oliveira LS, Sabourin R (2004) Support vector machines for handwritten numerical string recognition. In: Proceedings of the international workshop on frontiers in handwriting recognition (IWFHR-09), Tokyo, pp 39–44
16. Sadri J, Suen CY, Bui TD (2007) A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. Pattern Recogn 40(3):898–919
17. Kanungo T, Haralick RM (1990) Character recognition using mathematical morphology. In: Proceedings of USPS fourth advanced technology conference, vol 2. Washington, pp 973–986
18. Ha TM, Zimmermann M, Bunke H (1998) Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. Pattern Recogn 31(3):257–272
19. Elnagar A, Alhajj R (2003) Segmentation of connected handwritten numeral strings. Pattern Recogn 36(3):625–634
20. Kyung KK, Ho KJ, Ching YS (2002) Segmentation-based recognition of handwritten touching pairs of digits using structural features. Pattern Recogn Lett 23(1–3):13–24
21. Lei Y, Liu CS, Ding XQ, Fu Q (2004) A recognition based system for segmentation of touching handwritten numeral strings. In: Proceedings of 9th international workshop on frontiers in handwriting recognition (IWFHR-09), Tokyo, pp 294–299
22. Ma R, Zhao Y, Xia Y, Yan Y (2008) A touching pattern-oriented strategy for handwritten digits segmentation. In: Proceedings of the international conference on computational intelligence and security (CIS), vol 1. Suzhou, pp 174–179
23. Wang YJ, Liu XB, Jia YD (2009) Statistical modeling and learning for recognition-based handwritten numeral string segmentation. In: Proceedings of 10th international conference on document analysis and recognition (ICDAR-10), Barcelona, pp 421–425
24. Gattal A, Chibani Y (2015) SVM-based segmentation–verification of handwritten connected digits using the oriented sliding window. Int J Comput Intell Appl (IJCIA) 14(1):1–17
25. Terrades OR, Valveny E (2003) Radon transform for linear symbol representation. In: Proceedings of the seventh international conference on document analysis and recognition (ICDAR-03), vol 1. Edinburgh, pp 195–199
26. Manjunath AVN, Hemantha KG, Noushath S (2007) Robust unconstrained handwritten digit recognition using Radon transform. In: Proceedings of international conference on signal processing, communications and networking (ICSCN-07), Chennai, pp 626–629
27. Gattal A, Chibani Y, Djeddi C, Siddiqi I (2014) Improving isolated digit recognition using a combination of multiple features. In: Proceedings of 14th international conference on frontiers in handwriting recognition (ICFHR-2014), Crete Island, pp 446–451
28. Vellasques E, Oliveira LS, Britto AS, JrAL Koerich, Sabourin R (2008) Filtering segmentation cuts for digit string recognition. Pattern Recogn 41(10):3044–3053
29. Grother PJ (1995) NIST special database 19-handprinted forms and characters database. National Institute of Standards and Technology (NIST), Gaithersburg
30. Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999