Pattern Recognition 48 (2015) 4036–4048

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Junction detection in handwritten documents and its application to writer identification

Sheng He*, Marco Wiering, Lambert Schomaker

Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen, PO Box 407, 9700 AK Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 12 December 2014
Received in revised form
21 April 2015
Accepted 26 May 2015
Available online 4 June 2015

Keywords:
Handwriting recognition
Junction detection
Cross-script
Writer identification
Junclets

ABSTRACT

In this paper, we propose a novel junction detection method in handwritten images, which uses the stroke-length distribution in every direction around a reference point inside the ink of texts. Our proposed junction detection method is simple and efficient, and yields a junction feature in a natural manner, which can be considered as a local descriptor. We apply our proposed junction detector to writer identification by *Junclets* which is a codebook-based representation trained from the detected junctions. A new challenging data set which contains multiple scripts (English and Chinese) written by the same writers is introduced to evaluate the performance of the proposed junctions for cross-script writer identification. Furthermore, two other common data sets are used to evaluate our junction-based descriptor. Experimental results show that our proposed junction detector is stable under rotation and scale changes, and the performance of writer identification indicates that junctions are important atomic elements to characterize the writing styles. The proposed junction detector is applicable to both historical documents and modern handwritings, and can be used as well for junction retrieval.

1. Introduction

Singular structural features [1] are informative elements in visual patterns. Especially, where curvilinear lines form a cross, there exist small, informative areas. Such crossing regions, or junctions in this paper, are of primary importance for character perception and recognition. The junctions can be categorized into different types such as L-, T-(or Y-) and X-junctions [2] according to the number of edges they connect, or the number of branches they have. Fig. 1 shows several artificial junctions. Given a combination of them, people can easily recognize the corresponding character. For example, given the junction set {(a), (b), (c)} in Fig. 1, the character **'A'** will pop up in our brain. Similarly, the combination of {(d), (e)} results in the character **'F'**, putting the set {(d), (e), (f)} together will form the character **'E'**, {(e), (g)} will form character **'H'**, {(a), (h), (a)} will be character **'M'**, and the different arrangement {(h), (a), (h)} will be character **'W'**. From this example we can conclude that junctions are important atomic elements for some English characters, and such atomic elements are shared between different characters. For instance, the junction (e) in Fig. 1 is shared between **'H'**,**'E'** and **'F'**.

Junctions are also prevalent in handwritten scripts for languages that use the Roman alphabets, some of which have inherent junctions. Since Chinese characters are composed of line-drawing strokes, they naturally contain many junction points [3]. Characters in other scripts probably also contain junctions, such as Arabic characters [4]. Junctions are often the consequence of overwritten curved traces of handwriting, or are the consequence of connecting strokes between characters, as illustrated in Fig. 2. Junctions reflect the local geometrical and structural features around the singular, salient points in handwritten texts. Hence, it is natural to use junctions in handwritten document analysis. Liu et al. [5] have shown the efficiency of using fork points on the skeletons for Chinese character recognition. It has also been used to extract features for Arabic handwriting recognition (see the survey [6]).

In this study, we take the assumption that junction shapes are not guaranteed to be identical for different writers. Furthermore, even the same characters written in different historical periods contain different junction shapes. Generally, the differences are from three aspects: first, the length of branches of the junctions are variant. Second, the angles between each branch are also different between different writers or in different periods. Third, the type of junctions might be changed. We believe that those differences are caused by individual writing habits which can be considered as one type of biometric feature. Such features can be used for writer identification and historical document dating [7].

Based on the observations that junctions are prevalent in handwritten documents and they are different when generated by different writers as mentioned above, we propose an approach to detect junctions in handwritten documents and evaluate the

* Corresponding author. Tel.:+31 503637410.
E-mail addresses: heshengxgd@gmail.com (S. He), M.A.Wiering@rug.n (M. Wiering), L.Schomaker@ai.rug.nl (L. Schomaker).

http://dx.doi.org/10.1016/j.patcog.2015.05.022
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

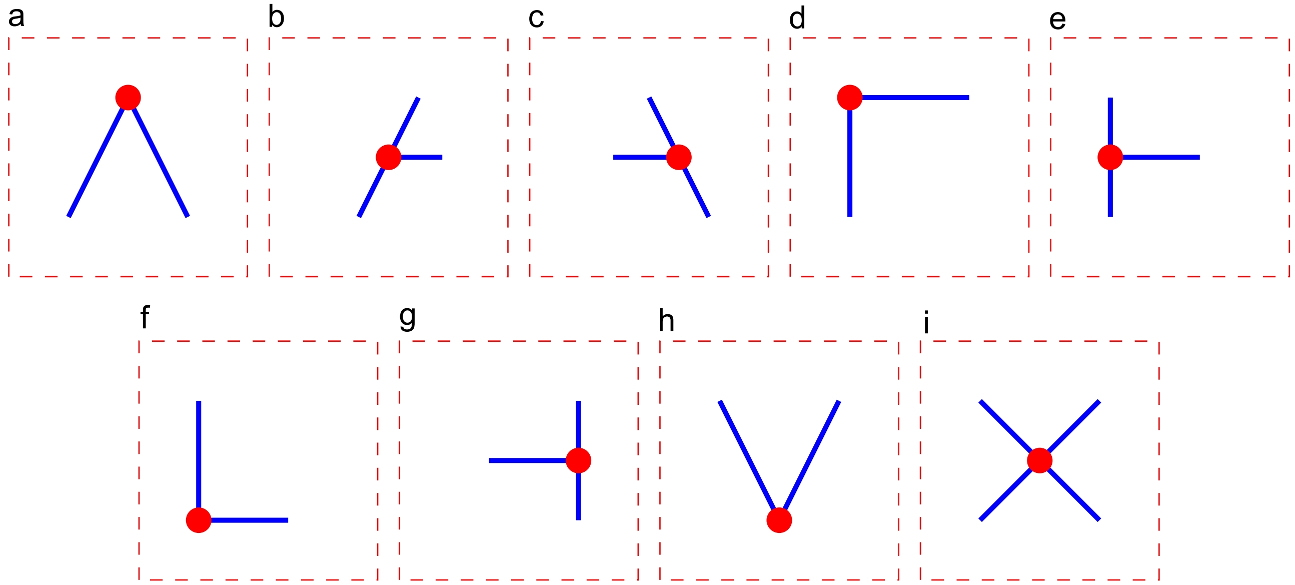

Fig. 1. Artificial junctions. The red dots are the center points of the junction, and the blue lines are the branches. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

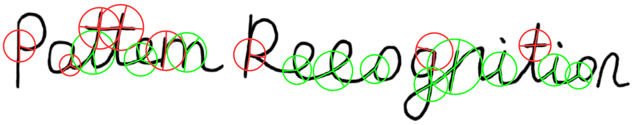

Fig. 2. Junctions in handwritten characters. The junctions marked in light (red) are within a character itself, and the green ones are from stroke touching. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

performances of using these detected junctions for writer identification. The contributions of this paper are summarized as follows: (1) we propose a simple yet effective method for junction detection in handwritten documents. (2) Our junction detector yields a junction feature, which can be considered as a mid-level feature representation. Furthermore, a new representation of handwritten documents is proposed based on the detected junctions, termed as *Junclets*, which are the primitive junctions of the document. The main advantage of the proposed method compared to junction detection in line-drawing images in [8] is that our proposed junction method can yield a junction feature in a natural manner. In addition, the benefit of the proposed *Junclets* compared to COnnected-COmponent COntour (CO^3) [9] and Fraglets [8] is that it does not rely on any segmentation or line detection which are challenging problems in document images, especially in historical documents where a connected component may span several lines due to touching ascenders and descenders.

This paper is organized as follows: we review the related work for junction detection and writer identification in Section 2. The third section presents the details of the proposed method. The writer identification method based on Junclets is described in Section 4. The experimental results are given in Section 5. We give the conclusion and future work in the last section.

2. Related work

2.1. Junction detection

In natural images, junctions are often detected based on template matching, contours, or gradient distributions. The template-based method for junction detection has been proposed in [2], in which the junction detection problem is formulated as one of finding the parameter values of the junctions that yield a junction which best approximates the template data by minimizing an energy function. The energy function has two parts: scale and location of junctions in images and the junction parameters, which are the number of wedges, wedge angles and wedge intensities. In [10], a novel junction detector is proposed by fitting the neighborhood around a point to a junction model, which segments the neighborhood into wedges by determining a set of radial edges. Two energy functions are used for radial segmentation, and junctions with the most energy are selected as junction candidates, followed by junction refinement to suppress the junctions on the straight edges. The contour-based approach [11] considers junctions as points at which two or more distinct contours intersect, and junctions are localized based on the combination of local and global contours using the *global probability of boundary* (gPb) [12]. Finally, a probability of junction operator is designed to compare the keypoints found by junctions to those detected by the Harris operator. Recently, Xia et al. [13] introduced a novel meaningful junction detection method based on the *a contrario* detection theory, called *a contrario* junction detection (ACJ). The strength of a junction is defined as the minimum of the branch strengths which is a measurement of the consistency of the gradient directions in an angular sector. Junctions are detected whose strength is greater than a threshold which is estimated by the *a contrario* approach. Compared to other methods, this approach requires fewer parameters, and is able to inhibit junctions in textured areas.

In [8], junctions are computed by searching for optimal meeting points of median lines in line-drawing images. There are three main steps in this method: (1) region of support determination by the linear least squares for 2-junctions, and crossing-points in skeleton lines for n-junctions, where $n=3$, 4. (2) Distorted zone construction by a circle centered at candidate junction points whose diameter is equal to the local line thickness. (3) Extracting the local topology which is a set of skeleton segments linked with a connected component distorted zone and junction optimization.

Su et al. [14] propose a method for junction detection in 2D images with linear structures. The Hessian information and correlation matrix measurements are combined to select the candidate junction points. The potential junction branches of candidate junctions are found, based on the idea that the linear structure should have a higher intensity compared to the background in structured images. Then the locations of the junction centers are refined using template fitting at multiple scales. One disadvantage of this method is that it can only detect junctions with three or more branches.

2.2. *Writer identification*

Approaches to writer identification can be coarsely divided into text-dependent and text-independent groups, according to the criteria whether the method recognizes the individual writing style based on certain characters or words (text-dependent) or features extracted from the entire image regardless of the semantic content (text-independent). The text-dependent approaches are limited due to the facts that it requires text segmentation and recognition prior to writer recognition, and the examined characters, such as ‘d’, ‘y’ and ‘f’ in [15], should be present in the writing samples to be compared. In addition, those methods are unable to seize the writing styles across different characters. Therefore, many automated writer identification methods fall into the text-independent category, in which statistical features are extracted from the entire image of a text block, and the similarity between two pieces of text is obtained based on those extracted features.

The features used in text-independent approaches have typically been categorized into two classes: statistical features and codebook-based features. Several widely used statistical features have been proposed in the last two decades. In [16], the edge-based directional probability distribution, and the joint probability distribution of the angle combination of two “hinged” edge fragments are proposed for writer identification, which is termed as the “edge-Hinge” feature. This method has been extended to the contour-Hinge probability distribution [17] which computes a Hinge kernel on the contours of texts, Quill–Hinge [18] which combines the ink width with the contour-Hinge feature, and Δ^n*Hinge* [19] which is a rotation-invariant feature based on contour-Hinge, but incorporates the derivative between several points along the ink contours. Some methods use a filtering approach to extract features from text blocks, such as Gabor filtering [20,21], XGabor filter [22] and oriented Basic Image Features (oBIF) [23]. Chain codes and polygon based features on contours have also been used for writer identification [24]. Other features of writer identification in Indic scripts have been proposed, such as the autoregressive (AR) [25], gradient [26] or curvature [27] features and the radon transform projection profile [28].

The codebook-based features are inspired by the bag-of-visual-words framework [29] used in computer vision, which is useful in the case that some local elements are extracted from the images, but they can not be directly used to compare the similarity between two images. A codebook is learned from the local elements extracted from the entire data set in order to capture the general information. Finally, the feature vectors can be determined by computing the occurrence histogram of the members of the codewords in each image.

In writer identification, several local elements have been proposed to represent the handwritten text. In order to capture features of the pen-tip trajectory which contains valuable writer-specific information, Schomaker and Bulacu [9] considered the CO_3 as the basic elements sampling from the connected contours. Furthermore, this approach is extended to graphemes [17,4] which are the ink-blob shapes generated by the writers, and an improved segmentation method has been proposed in [30]. Similar codes, such as curve fragment and line fragment codes, are proposed in [31] to construct the codebook for writer identification.

Small parts of handwritten text which do not carry any semantic information [24] or characters and symbols [32] are extracted as codes to train the codebook to characterize the writer of a given text sample. Recently, a grapheme codebook is constructed based on the Beta-elliptic model for writer identification and verification in [33], which is model driven without training.

Theoretically, our proposed writer identification method using junctions of the characters does not fall into text-dependent or text-independent categories, according to the above analysis. Our method can be considered as a Semi-text dependent approach, because some English letters have no junctions, such as ‘C’, ‘O’ or ‘S’ and other lower-case characters. However, in the real world, due to the writing habits, junctions can also be generated in no-junction characters. Therefore, our approach can still be regarded as a text-independent method.

3. Junction detection

In a handwritten image, a junction is defined as a structure $\mathcal{J}$ on the text strokes, with a center point and several separated branches, which can be formulated as [13]: $\left\{\mathbf{p}, r, \{\theta\}_{m=1}^{M}, S(\theta_m)\right\}$, $\mathbf{p}$ is the center point inside the ink, $r \in \mathcal{N}$ is the scale of the junction, $\{\theta_1, \ldots, \theta_M\}$ are the M branch directions around $\mathbf{p}$ which are corresponding to stroke directions, M is the order of the junction which is always set to 2, 3 or 4, corresponding to L, Y or X-junctions. $S(\theta_m)$ is the strength of the branch with direction θ_m. An example of a junction with three branches is shown in Fig. 3. In this paper, the discrete set $\mathcal{D}$ of possible directions θ_m is defined as

$$\mathcal{D} = \left\{2\pi k/N; k \in \{0, \ldots, N-1\}\right\} \tag{1}$$

Here, $N \in \mathcal{N}$ is the number of directions we considered, which is set to 360 in all experiments in this paper.

According to the above definition, there are three main procedures of the proposed junction detection approach. Firstly, the candidate center point $\mathbf{p}$ is detected. After that, the strength of a branch in every direction in the discrete set $\mathcal{D}$ is obtained. Finally, candidate branches are found on local maximum directions, followed with several junction refinement operators. More details will be presented in the following sections.

3.1. *Pre-processing*

The input of our method is a binary document image with skeleton lines. The Otsu thresholding algorithm [34] is applied in this paper, which is widely used for modern handwritings [17,18]. Other robust binarization methods, such as the AdOtsu [35] method or the method proposed in [36], could be used for degraded documents depending on the application. Any skeleton extraction method can be used, because we only use the skeleton line for junction candidates detection, not for feature computation.

3.2. *Detection of candidate junctions*

The junctions in handwritten scripts are generated by the crossings of strokes in handwritten documents (see Fig. 2), hence it is reasonable to select the fork points obtained from the skeletonization process as the candidate center points. A fork point which is always

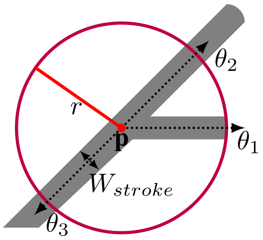

Fig. 3. A junction with three branches. Each branch lies on the skeleton line of the stroke.

related to a skeleton image is the location where at least three branches of line segments meet. Therefore, as mentioned in [8], this approach can only find the M-junctions ($M \geq 3$).

We use the method proposed in [18] to detect the 2-junction candidate center points. Given a point p_i on the skeleton line, two nearby pixels p_{i-e} and p_{i+e} can be found which have a distance of e pixels from p_i in preceding and succeeding directions. The leg from p_{i-e} to p_i forms an inbound angle ϕ_1, and the leg from p_i to p_{i+e} forms an outbound angle ϕ_2, as illustrated in Fig. 4(a). Then the angle on p_i defined by p_{i-e} and p_{i+e} can be estimated by: $\phi_{2\pi}(p_i) = \min(\|\phi_2 - \phi_1\|, 2\pi - \|\phi_2 - \phi_1\|)$. Fig. 4(b) shows the value $\phi_{2\pi}(p_i)$ of different positions from A to B on the line of Fig. 4(a). We can see that there is a local minimum where the curvature of the curve is higher (note that a small value corresponds to a high curvature). This point is considered as the candidate 2-junction center $\mathbf{p}$.

In order to make our 2-junction candidate detection method stable under scale changes, the parameter e should be adaptive. We found that the stroke width will change when the scale of the document image changes. Therefore, we set $e = \mu W_{stroke}$, in which W_{stroke} is the estimated half width of the stroke. The stroke width is estimated by the method proposed in [18], and the parameter μ is fixed as 1 in this paper.

Fig. 5 illustrates the candidate center points detected in an image. The detected 2-junction candidate center points, combined with fork points (i.e., points on skeleton lines with at least three 8-connected neighbors), are treated as the detected candidate junction center points. Finally, we randomly remove one of the points when they are close neighbors which is defined as the Manhattan distance of the center points is less than 4.

3.3. *Branch strength of a junction*

The strength of a junction branch $S(\theta)$ will be defined as a measurement of the probability of the direction θ being one of the branches of the junction. The branch strength is an important measurement for finding the potential branches. It has been computed in different ways for different types of images. For example, in linear structure images, the average intensity is used as the branch strength measurement in gray-scale images [14]. The consistency of a gradient distribution in a wedge region is used to find the potential branches in natural images [10,13]. In handwritten documents, junctions are always formed by the intersection of strokes. Therefore, it is natural to use the features of strokes to describe the strength in every direction.

The underlying idea for potential branch detection is that each branch should be one of the strokes which forms the junction, and the corresponding stroke length should be higher than the stroke lengths in neighboring directions. We can therefore consider the stroke length as the branch strength. There are some possible ways to compute such stroke lengths, such as searching the ink pixels following a ray in a certain direction, similar as [37]. In this paper, we use a simple and efficient method based on Bresenham's algorithm [38] to compute the length of the stroke inspired by [18].

Given a reference point (junction candidate center point) $\mathbf{p} = (x, y)$, one end point (x_e, y_e) is found in the direction θ by

$$x_e = x + l * \cos(\theta)$$
$$y_e = y + l * \sin(\theta) \tag{2}$$

Here, the parameter l signifies the maximum measurable length, and restricts the search space. The length of the stroke can be measured by the trace length on the Bresenham path [38] starting from reference point $\mathbf{p} = (x, y)$ towards the end point (x_e, y_e). The trace stops if a background (white) pixel (x_b, y_b) is hit and the trace length $len(\theta)$ is then computed as the distance from $\mathbf{p} = (x, y)$ to this background pixel (x_b, y_b) by the Euclidean measure:

$$len(\theta) = \sqrt{(x - x_b)^2 + (y - y_b)^2} \tag{3}$$

Fig. 6 gives an illustration of this method.

After computing the stroke length $len(\theta)$ at direction θ, the strength of a branch can be defined as

$$S(\theta) = len(\theta), \quad \theta \in \mathcal{D} \tag{4}$$

The larger the strength of a branch in direction θ, the more likely it is that the branch corresponding to one of the strokes formed the junction. Fig. 7 gives an illustration of the computed strength distribution in each direction in $\mathcal{D}$ given a reference point.

The strength of a branch computed by Eq. (4) is affected by the width of the stroke. The line with maximum length of the stroke is not parallel with the direction of the stroke, but shifted to the diagonal line, see Fig. 8(a). This fact results in two local maxima around the stroke direction. One possible solution is to use a smoothing filter to remove the noise, which is applied in [14]. However, designing such filter is difficult because it is related to the width of the stroke.

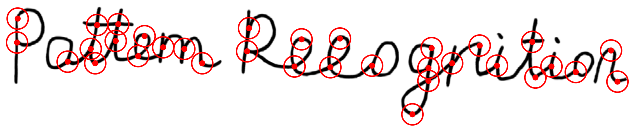

Fig. 5. Junction candidates in Fig. 2. The red points are the junction candidate center points. The red circles around the red points have no special meanings but improve visualization. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

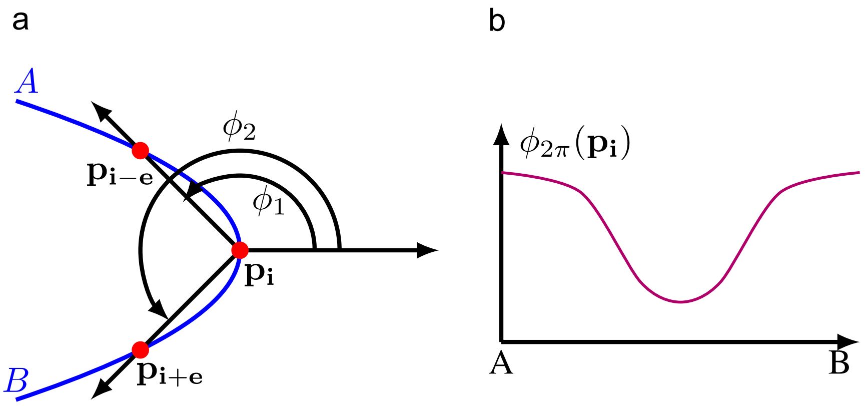

Fig. 4. Computing the angle of a curve. Figure (a) shows how to compute the angle on point p_i and (b) shows the computed angle on the curve from A to B.

One observation is that the direction of the stroke line lying on the maximal value of the distance transform (DT) map [39] of the binary stroke is the same as the direction of the stroke, see Fig. 8 (c). Based on this observation, the strength of a branch is weighted by the summed value of the distance transform $v_{dt}(\theta)$ in direction θ on the Bresenham path. The value v_{dt} in each direction is normalized by dividing the sum of $v_{dt}(\theta)$ among all the directions in $\mathcal{D}$:

$$S(\theta) = w(\theta) * len(\theta), \quad \theta \in \mathcal{D} \tag{5}$$

Here, $w(\theta) = v_{dt}(\theta) / \sum_{\theta=0}^{2\pi} v_{dt}(\theta)$. The local maximum point of the weighted strength reflects the direction of the stroke, see Fig. 8(d).

Based on the strength of a junction defined as Eq. (5), a fast and simple algorithm for junction detection can be developed. Given a point **p** and the strength $S(\theta), \theta \in \mathcal{D}$, the potential branches can be found in direction θ where the strength function $S(\theta)$ reaches a local maximum. The set of these local maxima can be computed efficiently by non-maximum suppression (NMS) [40,13].

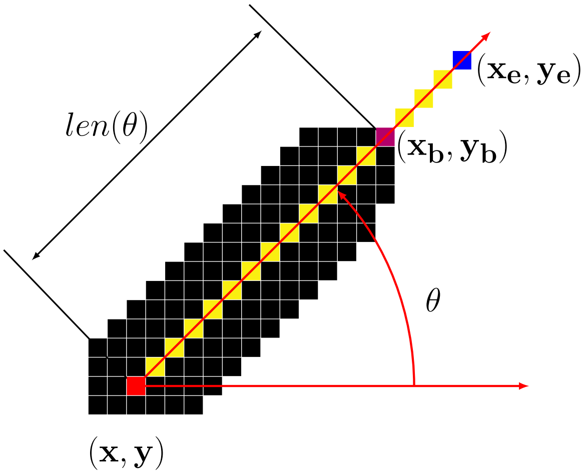
$len(\theta)$
$(\mathbf{x_e}, \mathbf{y_e})$
$(\mathbf{x_b}, \mathbf{y_b})$
θ
$(\mathbf{x}, \mathbf{y})$

Fig. 6. Illustration of length computation. The red point is the reference point (junction candidate center point), and the blue one is the end point with direction θ. The pink point is the first background pixel that is hit when following a Bresenham path from the start point (the red one) to the end point (the blue one). The length of the stroke on direction θ is the distance from the red point to the pink point. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

3.4. Final junction refinement

Sources of noise are easily introduced in the binarization, skeletonization or other pre-processing operations (see Fig. 10). Therefore, some post-processing steps are needed to refine the detected junctions.

(1) Remove the branches with short lengths. A branch whose length is less than a certain threshold is discarded, $len(\theta) < \lambda W_{stroke}$. Here, λ is a parameter to control the minimum of the length of the branches. For example, the green branch in Fig. 10(a) can be considered as noise and will be removed by this refinement.
(2) Remove overlapping branches. If the distance of two directions of branches $d_{2\pi} = |\theta_i - \theta_j|$ is smaller than Δ, the branch with smaller branch length $len(\theta)$ is discarded. For example, the green branch in Fig. 10(b) is too closed to the other branch, and will be removed.
(3) Suppress the junction on a straight line. We remove the 2-junctions whose branches are opposite $(d_{2\pi}(\theta_1, \theta_2 + \pi) < \Delta)$. Fig. 10(c) shows an example of a junction that lies on a straight line and will be removed by this constraint.

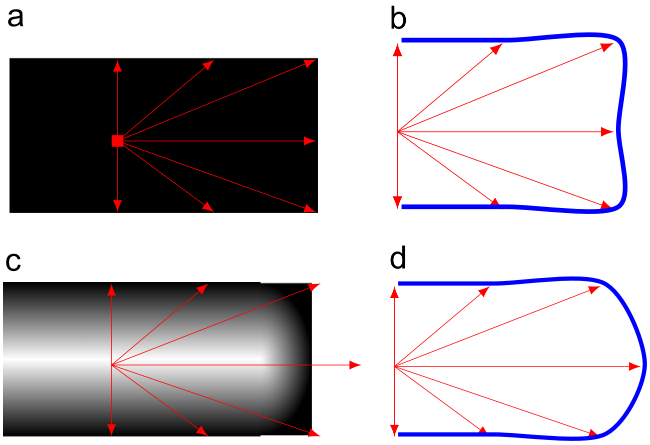

Fig. 8. Figure (b) is the strength of branches on figure (a), in which there are two local maximum points on the diagonal directions. Figure (d) shows the weighted strength of branches, in which there is only one maximum point on the stroke direction.

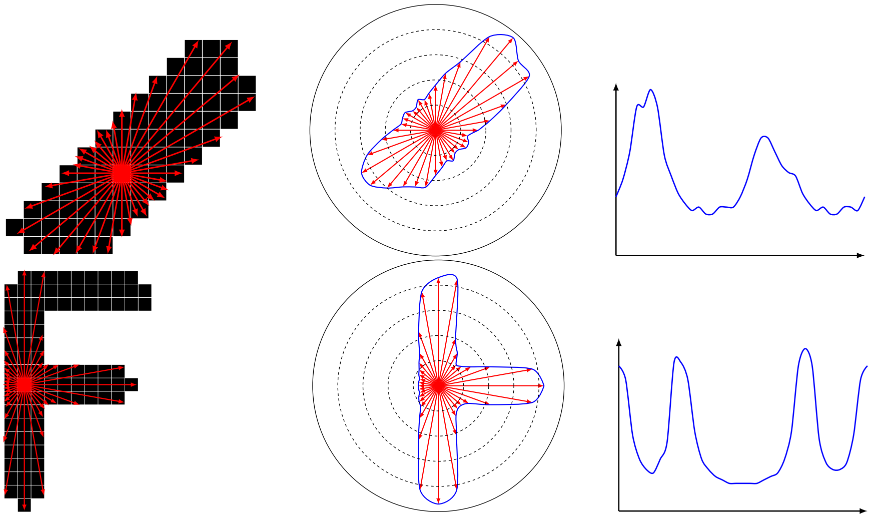

Fig. 7. An illustration of junction points and the strength of the branch in each direction. The middle figures show the length distribution in polar coordinates, and the right figures show the distribution in a linear coordinate, from 0 to 360°.

There are two parameters to refine the junctions: λ which controls the minimum length of branches and Δ which controls the overlapping and straight lines of the branches. The minimum value of $len(\theta)$ is equal to W_{stroke}, hence λ should be greater than 1. However, if λ is too large, meaningful branches will be deleted. Fig. 9 shows the detected percentages of different types of junctions with different parameter values of λ and Δ in the CERUG-EN data set (see Data Set section). We suggest to use $\lambda \in [1,2]$, and in the experiments we fixed it as $\lambda = 1.5$. For the parameter Δ, we suggest the value $\Delta = 0.1\pi$ because it is quite stable in the range $[0.02\pi, 0.2\pi]$.

The last issue is the scale r of the detected junction. Unlike other works [10,13,8,14], in our approach the scale r is not involved in the procedures of junction detection. In order to make the method complete, we just set the scale of the junction as the minimum of the length of the branches.

$$r = \min\{len(\theta_m)\}. \tag{6}$$

3.5. Junction feature

Template-based methods for junction detection involve similarity computation between the candidate junctions and the templates [14]. Building an efficient similarity measurement is a challenging problem. Two types of junction descriptors were proposed in [41] based on shape context approaches [42] for object recognition. In that paper the method concatenates the shape context features equally sampled on the contour segments of junctions. In this paper, we consider the normalized distribution of the stroke-length in each direction in $\mathcal{D}$ as a feature for the junction $\mathcal{J}_i$, which can be defined as

$$F(\mathcal{J}_i) = \{f_0, \ldots, f_{N-1}\} \tag{7}$$

Here, $f_i = len(\theta_i) / \sum_{j=0}^{N-1} len(\theta_j)$ is the normalized length of direction θ and $N=360$. The dimension of the feature is equal to N, which is the number of directions we considered. The last column of Fig. 7 gives two examples of the junction features.

There are several advantages of the proposed junction feature:

(1) It is a scale-invariant descriptor, which measures the context of a reference point inside the ink of texts. It is also easy to extend to a rotation-invariant descriptor by permuting the feature vector starting from some estimated angles, instead of horizontal directions.

(2) It contains the normalized ink width of the junction J_i, which can be estimated as

$$w_{stroke} \approx 2f_{\theta_{min}} \tag{8}$$

Here, $\theta_{min} = \text{argmin}_i\{f_i\}$ is the minimum value of the junction feature. The ink width has been shown as a powerful source of information for stroke determination [23].

(3) It also contains the normalized ink length on each branch direction of the junction.

(4) To the best of our knowledge, it is the first local descriptor in handwritten document analysis which can be used for matching, and recognition like SIFT [43] does in natural images.

3.6. Computational complexity

In this section, we provide a detailed complexity analysis of the junction detector on an input document image with size $w \times h$. In order to compute the skeleton points, a binarization method is applied first. In this paper, we use the Otsu binarization method, which requires one scan of the image to compute the histogram of gray levels and three scans from 0 to 255 to search the best

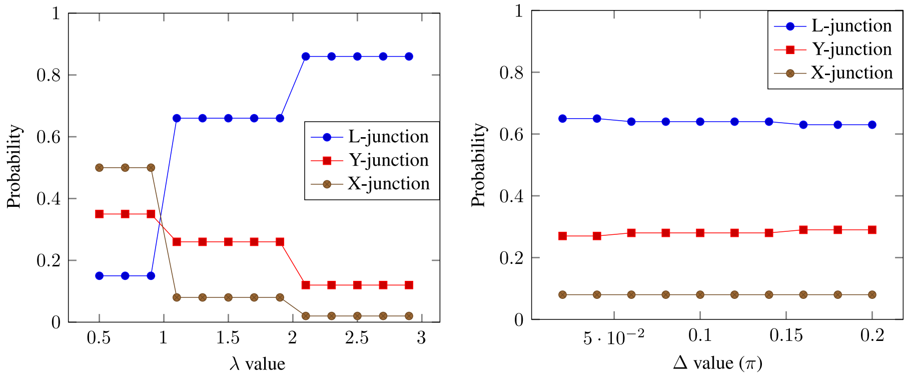

Fig. 9. The percentages of different types of junctions with different parameter values on CERUG-EN data set.

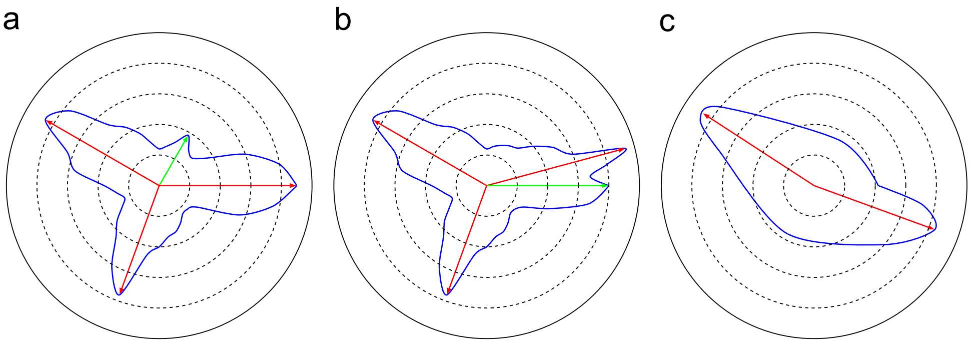

Fig. 10. Several detected junctions with noise. Figure (a) shows a junction with a branch whose length is short (the green branch), figure (b) shows a junction with overlapping branches and the green branch should be removed, and figure (c) shows a junction on an approximately straight line which should be removed. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

threshold. In the worst-case scenario, two scans of the image are required to calculate the medial axis. Generally, two scans are also needed to compute the distance map. In summary, the computational complexity of the binarization, skeletonization and distance map computation steps is $O(5wh)$, which is basically linear.

We assume that S skeleton points are obtained which is far less than $w \times h$. In order to search the high-curvature points, $2e$ points need to be checked for each skeleton point to compute the curvature of the skeleton line and one scan is needed to find the minimum value. In this paper, we set e equal to the stroke width, which is less than 20 in most cases for the modern handwritings. Therefore, the complexity of structure point detection is $O((2e+1)S)$.

Assuming that B structure points are detected, where $B < S$. For each structure point in the worst-case scenario, the computation of the junction feature requires $B = N_o \times L_{max}$ binary tests, where N_o is the number of checked orientations and L_{max} is the max search length. In this paper, we find that $N_o = 360$ and $L_{max} = 80$ performs well. For each junction, one scan on N_o dimensions is needed to find the position of the branches. The complexity of the junction computation on all the structure points is linear (e.g. $O(B \times (N_o \times L_{max} + 1))$.

4. Writer identification

Although a wide variety of features, local or global, have been proposed in the literature to distinguish writing styles, they are finally transformed into global features based on the bag-of-visual-words framework. Using global features to represent handwritten samples is simple and efficient for computation.

We build the probability distribution of the junctions as a global feature for each writer based on a learned codebook, which we termed as *Junclets*, with a similar framework as the traditional approaches to build a probability distribution of local patterns, such as connected-component contours (CO_3) [9], graphemes [17], writing fragments [24], and line segment codes [31]. Compared to the existing methods, our *Junclets* do not need any segmentation, which makes our method more stable and universal for any type of documents. Our basic idea is that the ensemble of junctions can capture the junction details of the handwritings, which reflect the writing styles of the author.

For each data set, the training set is generated by the junctions extracted from one of the handwritings from each writer. The Kohonen SOM 2D method is used to train the junction codebook, which is widely used in other works [44]. We evaluate the performance with different sizes of the codebook in our experimental section. One trained codebook is shown in Fig. 11. The writer can be characterized by a stochastic pattern generator, producing a family of basic shapes, such as graphemes [17] or junctions in this paper. The individual shape emission probability is computed by building a histogram based on the trained codebook using the top 5 nearest codewords coding method inspired by LLC [45]. The writer descriptor is computed by normalizing this histogram to a probability distribution that sums to 1.

5. Experimental results

5.1. Data sets

We used the handwritten documents from the existing widely used data sets, such as Firemaker [46] and IAM [47], to evaluate the performance of the proposed method for writer identification.

In modern times, more and more people can use more than one language, hence writer identification based on different languages is a new challenging problem, which has been studied in [23] among Latin language, such as English, French, German and Greek. However, there is no research report about how the handwriting style is affected by different characters using different alphabets. For example, how the English text written by Chinese people is affected by the way of writing Chinese characters. In order to answer such questions, we collect a new data set which contains multiple scripts (English and Chinese) from the same Chinese writer, called the Chinese-English database of the University of Groningen (CERUG for short).

The CERUG data set contains handwritten documents collected from 105 Chinese subjects, predominantly students from China. Some of them live in China and the rest studies in the Netherlands. Every subject is required to write four different A4 pages, following the Firemaker data set. On page 1, they were asked to copy a text of two paragraphs in Chinese. On page 2, the subjects described certain topics they liked in their own words in Chinese. We term the subset containing those two pages as CERUG-CN, in which handwritten documents are written in Chinese. Page 3 contains English text copied from two paragraphs. We split this page into two sub pages, and each sub page contains one paragraph. This forms the subset termed as CERUG-EN. In page 4, the subjects were asked to copy some names of countries and cities both in English and Chinese in two paragraphs. We also split this page into two sub pages to form another subset, which is termed as CERUG-MIXED for short. Note that each sub page in CERUG-MIXED contains both English letters and Chinese characters. In all three subsets, there are two handwritten samples from each writer. All the documents were scanned at 300 dpi, 8 bits/pixel, gray-scale.

5.2. Junction analysis

In this section, we evaluate the performance of the proposed junction detection method on the CERUG data set. Generally, the coherence of the detection through some transformations, such as scaling and rotation, is investigated under some criterion. The *repeatability* criterion is widely used for performance characterization both in junction detection [8,13] and local keypoint detectors in computer vision [48]. In this paper, we follow the evaluation method proposed in [8,48].

The repeatability criterion signifies that the junctions detected in reference image I_{ref} should be repeated in their transformed images $I_{trans} = trans\{I\}$ with some small error level ϵ in location. The repeatability score is calculated as the ratio between the corresponding junction center points and the minimum total number of junction center points visible in both images. The correspondences are identified by checking whether two junctions from each image are matched. Each detected junction j_t in I_{trans} is matched with junction j_r in I_{ref} if their center points are close enough, in the sense that $\| \mathbf{p}_{j_t} - \mathbf{p}_{j_r} \|_2 < \epsilon$. In this paper, we report the average repeatability rate on each data set.

5.2.1. Evaluation of rotation change

In this experiment, we use rotation operators to compute the I_{trans} given the reference image I_{ref}. The error level is fixed at $\epsilon = 3$ pixels. Table 1 presents the performance of the proposed junction detection method on the three data sets. Our proposed method is theoretically rotation invariant, and the results show that the proposed method is quite robust to rotation changes in all three data sets.

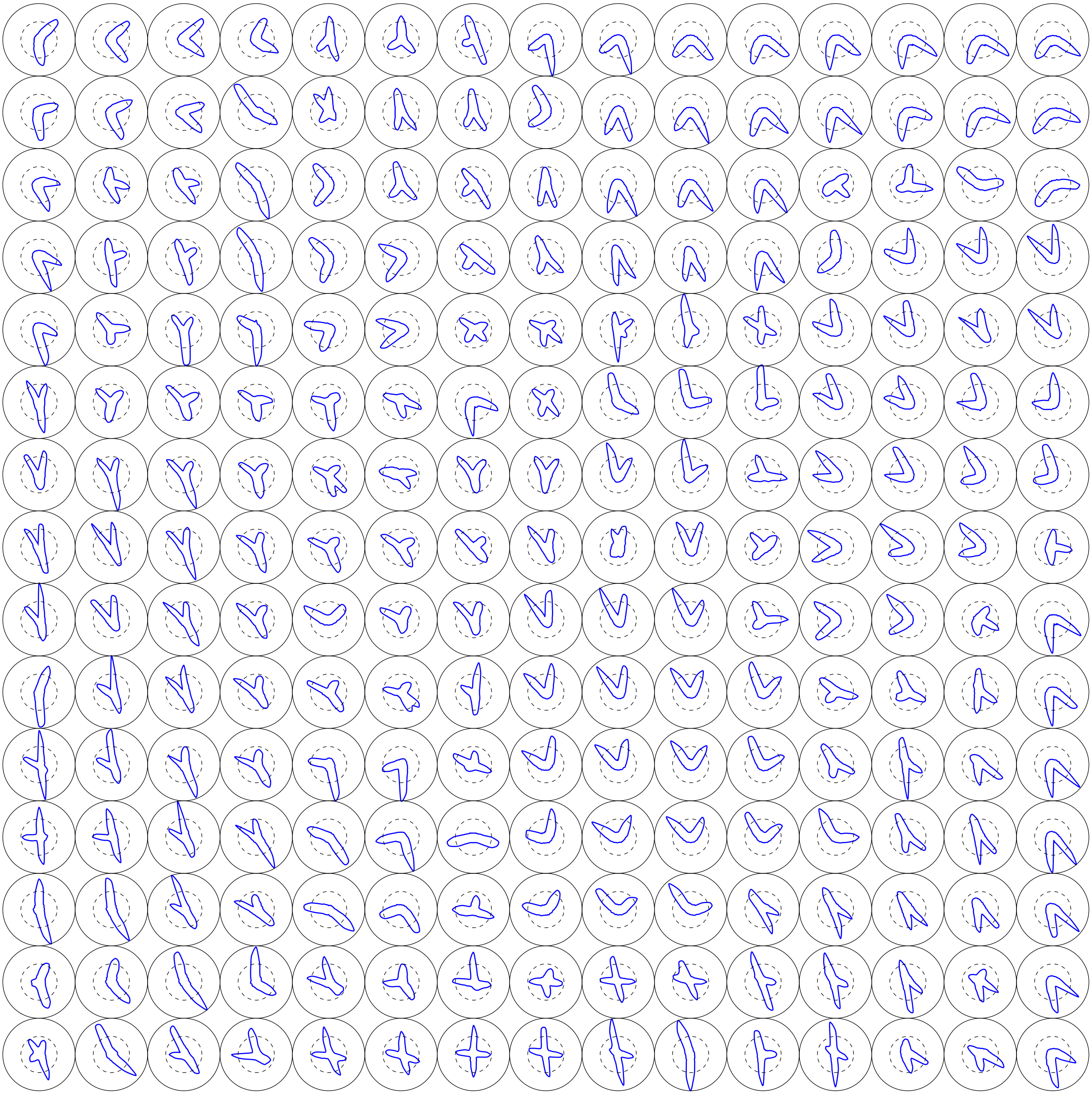

Fig. 11. Example of codebooks with 225 junctions. This codebook is trained using Kohonen 2D with a size of 15 × 15 on CERUG-EN (see data sets section).

Table 1
Repeatability rates on rotation change (%).

System	Rotation change IN (°)								
	10	20	30	40	50	60	70	80	90
CERUG-CN	80.8	78.8	77.8	77.3	76.8	76.4	77.1	78.1	81.2
CERUG-EN	79.5	76.5	74.6	74.1	73.5	73.4	74.3	76.1	80.3
CERUG-MIXED	79.8	77.5	76.2	75.7	75.2	74.9	75.7	76.6	80.1

5.2.2. Evaluation of scale change

In order to investigate the performance of the proposed junction detection through scale changes, we use the scale transform operators to obtain I_{trans} given the reference image I_{ref}. The set of scale factors is chosen as {0.3, 0.4, 0,5, 0.6, 0.7, 0.8, 0.9}. Table 2 shows the scale change results in our CERUG data sets, which demonstrate that our method is robust to scale changes. The repeatability rate always remains above 80%.

5.2.3. More visual results

This section aims at illustrating the proposed junction detection method with several more visual experiments on historical documents from the Monk system [49]. Fig. 12 presents the junction detection results. In these figures, the red circles and lines represent the junction region and the orientation of the branches respectively, and the white point in the center is the center point of the junction. Observe that our proposed method can accurately detect those junctions through their type, localization and scale in different types or layouts of handwritten historical scripts. We believe that the junctions are the basic elements or features which can be used for document and layout analysis and document classification.

5.3. *Performance of writer identification*

In this section, the performance of writer identification is presented using the detected junctions in CERUG, Firemaker and IAM data sets. We use the popular Top1 and Top10 identification rates to evaluate the performance of writer identification. Note that all the data sets contain two samples per writer and writer identification is performed in a “leave-one-out” manner. There are 105 writers in CERUG both in English and Chinese, 250 writers in Firemaker in Dutch, 650 writers in IAM in English.

5.3.1. *Performance of Junclets*

In this section, we conduct the experiment to evaluate the performance of writer identification with different sizes of the codebook. Fig. 13 shows our results obtained on different data sets. It can be noticed that the writer identification rates (Top1 and Top10) slightly increase as the size of the codebook increases. The Junclets codebook spans a shape space by providing a set of nearest-neighbor attractors for the junctions extracted from the written samples. As the codebook size increases, the Junclets in the codebook contain more detailed information, and therefore can capture more details of junctions generated by the individual writer. Furthermore, the dimension of the probability distribution is higher for larger codebook sizes, which results in a higher performance. However, from Fig. 13 we can find that the performance is slightly degraded when the codebook size is 2500. This is quite natural as a larger size of the codebook results in a larger dimensionality of the representation space which is more sensitive to the variance within the documents from the same writer. For the results reported in this paper, we used the codebook which contains 400 *Junclets*. Table 3 gives the writer identification performance on different data sets.

We also compare the results of our proposed Junclets features with Fraglets [17] which are computed by generating a codebook at the grapheme level. Table 4 gives the performance of these two methods based on two common data sets. The codebook size of Junclets is 400, which is equal to the one of Fraglets used in [17]. The results in Table 4 show that our proposed Junclets representation provides around 5% and 3% (Top1) better results than Fraglets on the Firemaker and IAM data sets, respectively.

Table 2
Repeatability rates on scale change (%).

System	Scaling factor						
	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CERUG-CN	86.9	81.8	81.4	81.3	82.0	82.3	82.5
CERUG-EN	83.4	80.9	82.3	83.6	84.1	83.9	83.4
CERUG-MIXED	85.8	82.2	81.4	81.8	82.2	82.2	82.2

In order to evaluate the Junclets feature with different sizes of training and testing data sets, we randomly select $\delta\%$ percentage of junctions in the texts both from training and testing samples from each writer. These selected junctions are then encoded in the

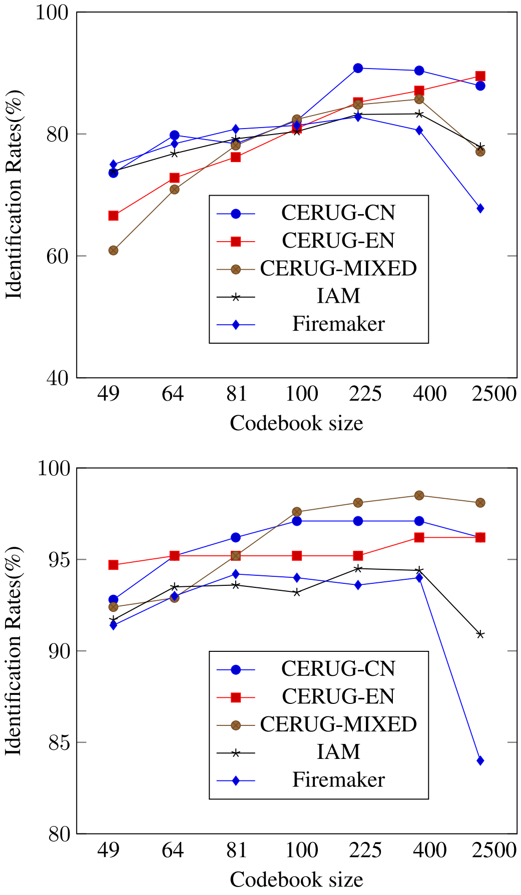

Fig. 13. The Top1 (top figure) and Top10 (bottom figure) performance of different sizes of the codebook on different data sets.

Table 3
Performance of Junclets (codebook size is 400) on different data sets.

Database	Top1	Top10
CERUG-CN	90.4	97.1
CERUG-EN	87.1	96.2
CERUG-MIXED	85.7	98.5
Firemaker	80.6	94.0
IAM	83.3	94.4

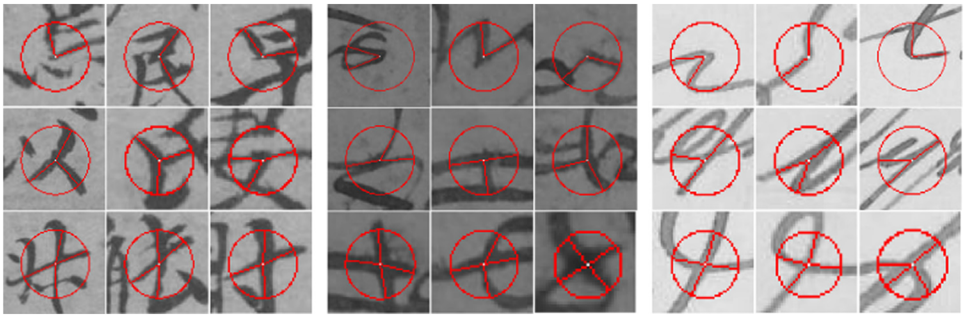

Fig. 12. Detected junctions in a historical document from Monk [49,50]. Note that all the junctions are normalized into a fixed size in order to improve visualization. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

trained Junclets codebook. This procedure is repeated 20 times, with a different random selection each time. The results are shown in Fig. 14. From the figure we can observe that the recognition rate is high with more training and testing junctions. This is because our proposed Junclets representation is a codebook-based statistical feature. More data leads to a more stable and robust feature.

5.3.2. *Performance of feature combinations*

To demonstrate the benefits of our proposed Junclets representation, we combine our method with other widely used features. In Table 5, we present the performance of other existing features and the combination with Junclets on the three data sets. The existing features we selected are:

(1) Hinge [17]: the joint probability distribution of the orientations of two legs of two contour fragments attached at a common end pixel on the contours.
(2) Quill [18]: the probability distribution of the relation between the ink direction and the ink width.
(3) QuillHinge [18]: the combined feature from Hinge and Quill. There are several common parameters in Hinge and Quill: the leg length r, the number of ink width bins p and the number of ink angle bins q. In their original works [17,18], those parameters were learned from the training data set. However, in this paper, we simply fixed them as $r=7, p=40, q=23$, because the aim is not to show the best performance of those features, but to provide the performance of them combined with Junclets.

Table 4
Performance comparison of Fraglets [17] and the proposed Junclets on writer identification.

Feature/Method	Firemaker		IAM	
	Top1	Top10	Top1	Top10
Fraglets [17]	75	92	80	94
Junclets	80.6	94.0	83.3	94.4

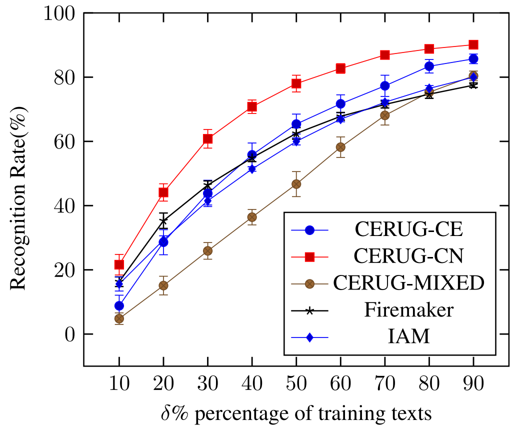

Fig. 14. The performance of the Junclets method on different size of the data sets. Error bars indicate the standard deviation of the performance across the 20 runs.

We use the weighted combination method between the considered feature and Junclets as: $d=(1-\lambda)d_c+\lambda d_{Junclets}$, where d_c is the distance of the considered features (one of Hinge, Quill and QuillHinge) and λ is the mixing coefficient. In our experiments, we empirically set $\lambda=0.2$ for Hinge, $\lambda=0.7$ for Quill, and $\lambda=0.6$ for QuillHinge. Here we use a high λ value for Quill than Hinge because our proposed Junclets representation also contains the stroke width information. Therefore, the impact of Quill is low when combined with the Junclets.

As shown in Table 5, the combination of the existing features and our proposed Junclets outperforms both the previous features and the Junclets representation on the three data sets. One interesting observation is that the edge-based features (Hinge, Quill and QuillHinge) do not achieve a good performance on the CERUG-EN data set compared to the Firemaker and IAM data sets. One partial reason is that English texts written by Chinese people have large straight lines compared to those written by native-speaker subjects. We performed an experiment to prove our assumption using a fast line detection method (LSD) [51] to detect lines in handwritten document in the Firemaker, IAM and CERUG-EN data sets. A histogram of the line length from 0 to 300 is built based on the detected lines, and the expectation of this empirical distribution is obtained. We also compute the integrated rate which is defined as the sum of the line length probability with the condition that the length is greater than a threshold T. We set $T=100$ in this experiment.

From Table 6 we can conclude that (1) the expectation of the line length on the three data sets is almost the same, which means the handwriting samples in those data sets are under the same scale. (2) The integrated probability of line lengths greater than $T=100$ of CERUG-EN is about 48 times and 8 times higher than the ones in Firemaker and IAM. The results demonstrate that the CERUG-EN is a challenging data set whose handwriting samples contain more long lines.

5.3.3. *Cross-script writer identification*

In this section, we look at the performance of writer identification between different scripts, especially between Chinese and English. We chose the first page of the CERUG-CN data set, and the first paragraph of the CERUG-EN and CERUG-MIXED data sets from each writer. The Junclets representation is computed based on the codebook trained from CERUG-MIXED because it contains both Chinese and English scripts. The performance of writer identification across different scripts is given in Table 7. These results show

Table 5
The performance of different features and the combination with Junclets.

Feature/Method	CERUG-CN		CERUG-EN		CERUG-MIXED		Firemaker		IAM	
	Top1	Top10	Top1	Top10	Top1	Top10	Top1	Top10	Top1	Top10
Hinge [17]	90.8	96.2	12.3	30.0	84.7	95.7	85.8	95.8	86.6	95.2
Quill [18]	82.7	92.3	18.5	48.6	74.8	93.3	60.8	78.8	84.6	93.8
QuillHinge [18]	88.5	93.8	45.2	91.0	86.7	98.6	74.0	89.8	90.8	96.5
Junclets	90.4	**97.1**	87.1	96.2	85.7	98.5	80.6	94.0	83.3	94.4
Junclets+Hinge	**94.2**	**97.1**	39.1	76.7	95.2	98.6	**89.8**	**96.0**	90.6	96.7
Junclets+Quill	92.3	96.2	86.2	97.1	92.9	**100**	83.4	95.0	89.4	96.5
Junclets+QuillHinge	92.3	95.2	**89.5**	**97.6**	**96.2**	**100**	85.2	95.4	**91.1**	**97.2**

that our proposed Junclets representation achieves much better results, particularly between Chinese and English scripts, on which the Hinge, Quill and QuillHinge fail in this test.

Table 6
The expectation and integrated rates of the line length distribution of three data sets in pixels.

Line-length	CERUG-EN	Firemaker	IAM
Expectation	20.3	19.6	19.0
Integrated rates ($L > 100$)	0.3933%	0.0082%	0.0479%

Table 7
The writer identification performance across scripts.

Feature/Method	Chinese/English		Chinese/Mixed		English/Mixed	
	Top1	Top10	Top1	Top10	Top1	Top10
Hinge [17]	0.48	0.96	11.0	38.8	3.8	16.2
Quill [18]	2.4	24.9	12.9	35.9	16.7	41.9
QuillHinge [18]	0.96	12.9	15.3	46.4	15.2	42.2
Junclets	**90.7**	**96.7**	**72.8**	**87.5**	**60.3**	**68.9**

Table 8
Writer identification performance of different approaches on the IAM and Firemaker datasets.

Approach	IAM			Firemaker		
	Writers	Top1	Top10	Writers	Top1	Top10
Wu et al. [52]	657	98.5	99.5	250	92.4	98.9
Siddiqi and Vincent [24]	650	89	97	–	–	–
Bulacu and Schomaker [17]	650	89	97	250	83	95
Ghiasi and Safabakhsh [31]	650	93.7	97.7	250	89.2	98.6
Jain and Doermann [53]	300	93.3	96.0	–	–	–
He and Schomaker [19]	650	93.2	97.2	250	90.4	98.2
Proposed	650	91.1	97.2	250	89.8	96.0

5.3.4. Comparison with other studies

We summarized the results of several works about writer identification on IAM and Firemaker data sets in the literature in Table 8. Although it is not fair to compare them because some approaches used a subset of the IAM database, Table 8 still gives us a good basis for comparison and our proposed method combined with other features is comparable with others. Although the writer identification rate of our method does not achieve the state-of-the-art results, the proposed Junclets representation can work on the challenging CERUG-EN data set and performs writer identification between Chinese and English.

5.4. Junction retrieval

We present here an additional experiment at application level for junction retrieval. Junction retrieval, which is similar to word-image retrieval [50], is defined as: given a query junction, the top of the sorted list is obtained based on a large collection of junction instances. Because there is no such data set about junction retrieval, in this section, we give visual results. Fig. 15 shows the sorted hit list of the query junction (first column of each row) from the four pages of one hand on the CERUG data set. Note that our proposed junction features can find their nearest neighbors, and some similar junctions appear in both Chinese and English scripts from the same writer, which shows the strong power of junctions in writer identification, and other applications.

6. Conclusion and future work

In this paper, we have introduced a generic approach for junction detection in handwritten documents. The proposed method yields a junction feature in a natural manner, which can be considered as a local descriptor. We apply the detected junctions to writer identification using a compact representation, called *Junclets*.

The proposed *Junclets* representation which is computed from a learned codebook achieves much better performance for writer identification, especially across English and Chinese scripts on our

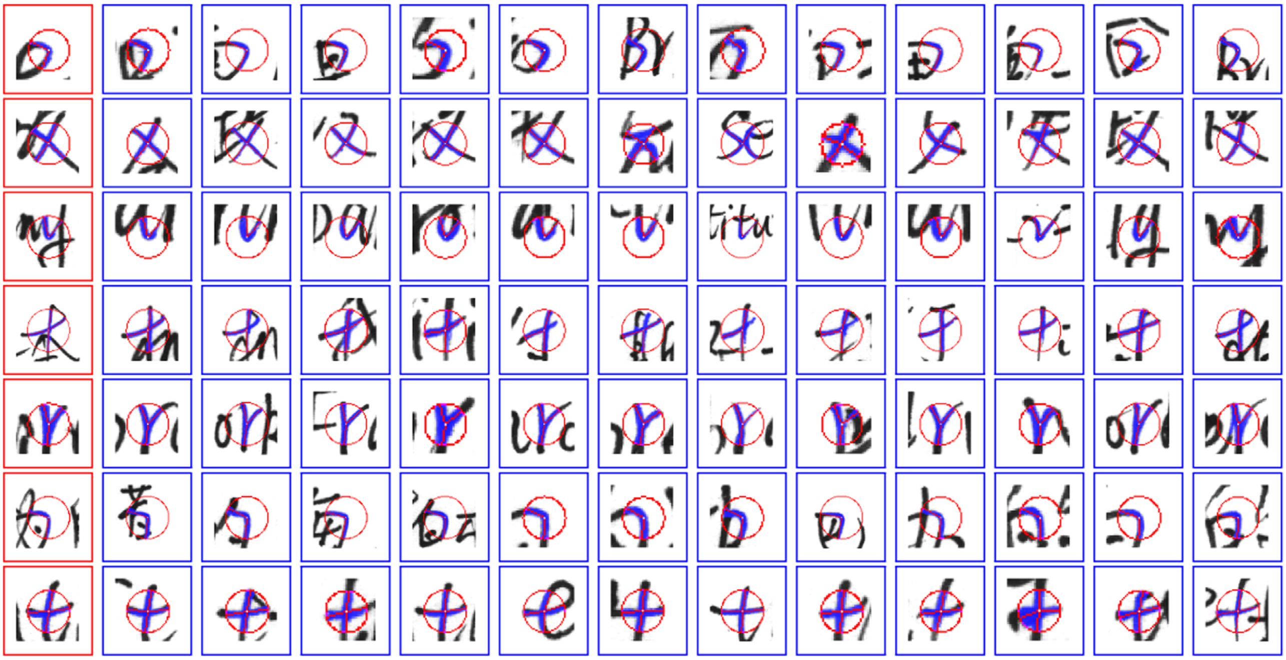

Fig. 15. Each row shows the first 13 instances in a hit list of the query junction (first column in the red box). The blue color shows the junction region on the text. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

novel data set. Our proposed method is simple and computationally efficient, and it does not rely on any segmentation, and hence can be used for any type of handwritten documents.

Junctions are very useful cues and features in handwritten document analysis. The junction features can be considered as a local descriptor for the reference junction center point. Therefore, this work opens several perspectives. Firstly, as a local feature, it can be directly used for word-spotting. Secondly, in our future work, junctions will be used for historical document dating [7], which aims at mining the temporally discriminative information contained in the text of historical documents. Thirdly, junctions only capture the information around a singular stroke point. They fail to describe the whole shape of strokes, which might be important for writer identification. In future work, we try to extend this method to compute the stroke features.

Conflict of interest

None declared.

Acknowledgements

This work has been supported by the Dutch Organization for Scientific Research NWO (Project no. 380-50-006).

References

[1] J.-C. Simon, O. Baret, Regularities and singularities in line pictures, Int. J. Pattern Recognit. Artif. Intell. 5 (1991) 57–77.

[2] L. Parida, D. Geiger, R. Hummel, Junctions: detection, classification, and reconstruction, IEEE Trans. Pattern Anal. Mach. Intell. 20 (7) (1998) 687–698.

[3] K.-C. Fan, W.-H. Wu, A run-length-coding-based approach to stroke extraction of Chinese characters, Pattern Recognit. 33 (11) (2000) 1881–1895.

[4] M. Bulacu, L. Schomaker, A. Brink, Text-independent writer identification and verification on offline Arabic handwriting, in: International Conference on Document Analysis and Recognition, vol. 2, 2007, pp. 769–773.

[5] K. Liu, Y.S. Huang, C.Y. Suen, Identification of fork points on the skeletons of handwritten Chinese characters, IEEE Trans. Pattern Anal. Mach. Intell. 21 (10) (1999) 1095–1100.

[6] L.M. Lorigo, V. Govindaraju, Offline Arabic handwriting recognition: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 28 (5) (2006) 712–724.

[7] S. He, P. Samara, J. Burgers, L. Schomaker, Towards style-based dating of historical documents, in: International Conference on Frontiers in Handwriting Recognition, 2014, pp. 265–270.

[8] T.-A. Pham, M. Delalandre, S. Barrat, J.-Y. Ramel, Accurate junction detection and characterization in line-drawing images, Pattern Recognit. 47 (1) (2014) 282–295.

[9] L. Schomaker, M. Bulacu, Automatic writer identification using connected-component contours and edge-based features of uppercase western script, IEEE Trans. Pattern Anal. Mach. Intell. 26 (4) (2004) 787–798.

[10] E.D. Sinzinger, A model-based approach to junction detection using radial energy, Pattern Recognit. 41 (2) (2008) 494–505.

[11] M. Maire, P. Arbeláez, C. Fowlkes, J. Malik, Using contours to detect and localize junctions in natural images, in: Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[12] D.R. Martin, C.C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.

[13] G.-S. Xia, J. Delon, Y. Gousseau, Accurate junction detection and characterization in natural images, Int. J. Comput. Vis. 106 (1) (2014) 31–56.

[14] R. Su, C. Sun, T.D. Pham, Junction detection for linear structures based on Hessian, correlation and shape information, Pattern Recognit. 45 (10) (2012) 3695–3706.

[15] V. Pervouchine, G. Leedham, Extraction and analysis of forensic document examiner features used for writer identification, Pattern Recognit. 40 (3) (2007) 1004–1013.

[16] M. Bulacu, L. Schomaker, Writer style from oriented edge fragments, Computer Analysis of Images and Patterns, Springer, Berlin, 2003, pp. 460–469.

[17] M. Bulacu, L. Schomaker, Text-independent writer identification and verification using textural and allographic features, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 701–717.

[18] A. Brink, J. Smit, M. Bulacu, L. Schomaker, Writer identification using directional ink-trace width measurements, Pattern Recognit. 45 (1) (2012) 162–171.

[19] S. He, L. Schomaker, Delta-n hinge: rotation-invariant features for writer identification, in: International Conference on Pattern Recognition, 2014, pp. 2023–2028.

[20] H.E. Said, T.N. Tan, K.D. Baker, Personal identification based on handwriting, Pattern Recognit. 33 (1) (2000) 149–160.

[21] F. Shababi, M. Rahmati, A new method for writer identification of handwritten Farsi documents, in: International Conference on Document Analysis and Recognition, 2009, pp. 426–430.

[22] B. Helli, M.E. Moghaddam, A text-independent Persian writer identification based on feature relation graph (FRG), Pattern Recognit. 43 (6) (2010) 2199–2209.

[23] A.J. Newell, L.D. Griffin, Writer identification using oriented basic image features and the delta encoding, Pattern Recognit. 47 (6) (2014) 2255–2265.

[24] I. Siddiqi, N. Vincent, Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features, Pattern Recognit. 43 (11) (2010) 3853–3865.

[25] U. Garain, T. Paquet, Off-line multi-script writer identification using AR coefficients, in: International Conference on Document Analysis and Recognition, 2009, pp. 991–995.

[26] S. Chanda, K. Franke, U. Pal, T. Wakabayashi, Text independent writer identification for Bengali script, in: International Conference on Pattern Recognition, 2010, pp. 2005–2008.

[27] S. Chanda, K. Franke, U. Pal, Text independent writer identification for Oriya script, Int. Worksh. Doc. Anal. Syst. (2012) 369–373.

[28] S. Biswas, A.K. Das, Writer identification of Bangla handwritings by radon transform projection profile, Int. Worksh. Doc. Anal. Syst. (2012) 215–219.

[29] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 524–531.

[30] G. Ghiasi, R. Safabakhsh, An efficient method for offline text independent writer identification, in: International Conference on Pattern Recognition, 2010, pp. 1245–1248.

[31] G. Ghiasi, R. Safabakhsh, Offline text-independent writer identification using codebook and efficient code extraction methods, Image Vis. Comput. 31 (5) (2013) 379–391.

[32] S. Marinai, B. Miotti, G. Soda, Bag of characters and SOM clustering for script recognition and writer identification, in: International Conference on Pattern Recognition, 2010, pp. 2182–2185.

[33] N.A. Mohamed, K. Maher, A model-based approach to offline text-independent Arabic writer identification and verification, Pattern Recognit. 48 (5) (2015) 1890–1903.

[34] N. Otsu, A threshold selection method from gray-level histograms, Automatica 11 (286–296) (1975) 23–27.

[35] R.F. Moghaddam, M. Cheriet, Adotsu: an adaptive and parameterless generalization of Otsu's method for document image binarization, Pattern Recognit. 45 (6) (2012) 2419–2431.

[36] B. Su, S. Lu, C.L. Tan, Robust document image binarization technique for degraded document images, IEEE Trans. Image Process. 22 (4) (2013) 1408–1417.

[37] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 2963–2970.

[38] D. Hearn, M.P. Baker, C version, Computer Graphics, 2, Prentice Hall, Upper Saddle River, 1997.

[39] A. Meijster, J.B. Roerdink, W.H. Hesselink, A general algorithm for computing distance transforms in linear time, Mathematical Morphology and its applications to image and signal processing, Springer, Berlin, 2000, pp. 331–340.

[40] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, in: Conference on Pattern Recognition, vol. 3, 2006, pp. 850–855.

[41] B. Wang, X. Bai, X. Wang, W. Liu, Z. Tu, Object recognition using junctions, in: Computer Vision–ECCV 2010, Springer, Berlin, 2010, pp. 15–28.

[42] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.

[43] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[44] L. Schomaker, K. Franke, M. Bulacu, Using codebooks of fragmented connected-component contours in forensic and historic writer identification, Pattern Recognit. Lett. 28 (6) (2007) 719–727.

[45] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[46] L. Schomaker, L. Vuurpijl, Forensic Writer Identification: a Benchmark Data Set and a Comparison of Two Systems, Technical Report, NICI, Nijmegen, 2000.

[47] U.-V. Marti, H. Bunke, The IAM-database: an English sentence database for offline handwriting recognition, Int. J. Doc. Anal. Recognit. 5 (1) (2002) 39–46.

[48] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. van Gool, A comparison of affine region detectors, Int. J. Comput. Vis. 65 (1–2) (2005) 43–72.

[49] T. van der Zant, L. Schomaker, K. Haak, Handwritten-word spotting using biologically inspired features, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1945–1957.

[50] J.-P. van Oosten, L. Schomaker, Separability versus prototypicality in handwritten word-image retrieval, Pattern Recognit. 47 (3) (2014) 1031–1038.

[51] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: a fast line segment detector with a false detection control, IEEE Trans. Pattern Anal. Mach. Intell. 32 (4) (2010) 722–732.

[52] X. Wu, Y. Tang, W. Bu, Offline text-independent writer identification based on scale invariant feature transform, IEEE Trans. Inf. Forens. Secur. 9 (3) (2014) 526–536.

[53] R. Jain, D. Doermann, Offline writer identification using k-adjacent segments, in: Conference on Document Analysis and Recognition, 2011, pp. 769–773.

Sheng He received the M.Sc. Degree in pattern recognition and intelligent system from the Northwestern Polytechnical University, Xi'an, China, in 2012. He is currently a Ph.D. student at the Institute of Artificial Intelligence and Cognitive Engineering in the University of Groningen, the Netherlands.

Marco Wiering studied computer science at the University of Amsterdam and finished Cum Laude in 1995. Then he went to IDSIA, a research institute in Lugano, Switzerland to do his Ph.D. studies. After he finished in 1999 with a Ph.D. thesis on Reinforcement Learning, he became a post-doc for several months at the University of Amsterdam. In 2000 he became an assistant professor at Utrecht University in the Intelligent Systems Group from the Institute of Information and Computing Sciences. In 2007 he became an assistant professor with tenure track in the Institute of Artificial Intelligence and Cognitive Engineering at the University of Groningen. He is currently an associate editor for IEEE Transactions on Neural Networks and Learning Systems. He has (co-)authored more than 100 journal and conference papers on topics such as reinforcement learning, machine learning, computer vision, pattern recognition, and robotics.

Lambert Schomaker is a professor of artificial intelligence at the University of Groningen, the Netherlands. He has produced over 140 publications, predominantly in pattern recognition, is member of IEEE and IAPR, and received the IBM Faculty Awards (2011, 2012) for the Monk word retrieval system in historical manuscript collections using high-performance computing.