IJDAR (2017) 20:79–89
DOI 10.1007/s10032-017-0281-y

ORIGINAL PAPER

Online recognition of handwritten music symbols

Jiyong Oh[1] · Sung Joon Son[2] · Sangkuk Lee[2] · Ji-Won Kwon[3] · Nojun Kwak[2]

Received: 28 July 2016 / Revised: 5 December 2016 / Accepted: 8 February 2017 / Published online: 15 February 2017

Abstract In this paper, we propose an effective online method to recognize handwritten music symbols. Based on the fact that most music symbols can be regarded as combinations of several basic strokes, the proposed method first classifies all the strokes comprising an input symbol and then recognizes the symbol based on the results of stroke classification. For stroke classification, we propose to use three types of features, which are the size information, the histogram of directional movement angles, and the histogram of undirected movement angles. When combining classified strokes into a music symbol, we utilize their sizes and spatial relation together with their combination. The proposed method is evaluated using two datasets including HOMUS, one of the largest music symbol datasets. As a result, it achieves a significant improvements of about 10% in recognition rates compared to the state-of-the-art method for the datasets. This shows the superiority of the proposed method in online handwritten music symbol recognition.

✉ Nojun Kwak
nojunk@snu.ac.kr

Jiyong Oh
jiyongoh@etri.re.kr

Sung Joon Son
sjson718@snu.ac.kr

Sangkuk Lee
sangkuklee@snu.ac.kr

Ji-Won Kwon
tonkunst@suwon.ac.kr

[1] Daegu-Gyeongbuk Research Center, Electronics and Telecommunications Research Institute, Daegu, Korea

[2] Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea

[3] Department of Composition, The University of Suwon, Hwaseong, Korea

Keywords Music symbol recognition · Online handwritten symbol recognition · Histogram of directional movement angles · Histogram of undirected movement angles · Spatial relation

1 Introduction

As information technology has been dramatically advanced during the last two decades, digitization deeply invades various fields of music such as preservation, duplication and distribution [1], and many composers and songwriters use digital devices and computer softwares nowadays. Although those programs provide the function of making and editing music scores using music instruments and computer input devices, pen and paper still occupy an important position among various composition tools. Hence, automatic recognition of handwritten music symbols has been required. Especially, its demand increases more and more as pen-based digital devices such as smartphones and tablet PCs are widely used.

Music symbol recognition methods can be divided into two categories. The techniques belonging to the first category are referred to as the optical music recognition (OMR) or the offline methods, where music symbols in score are recognized in the form of images. Many works have been presented for OMR and their details can be found in [2]. It is known that the main challenges of the methods are to remove staff from music score and to segment each meaningful symbol. On the other hand, the techniques belonging to the second category are called the online methods because music symbols can be recognized by those methods right after they have been written. Different from OMR methods, they can utilize time information of written symbols. This makes each symbol to be relatively easily isolated in the online methods. To utilize

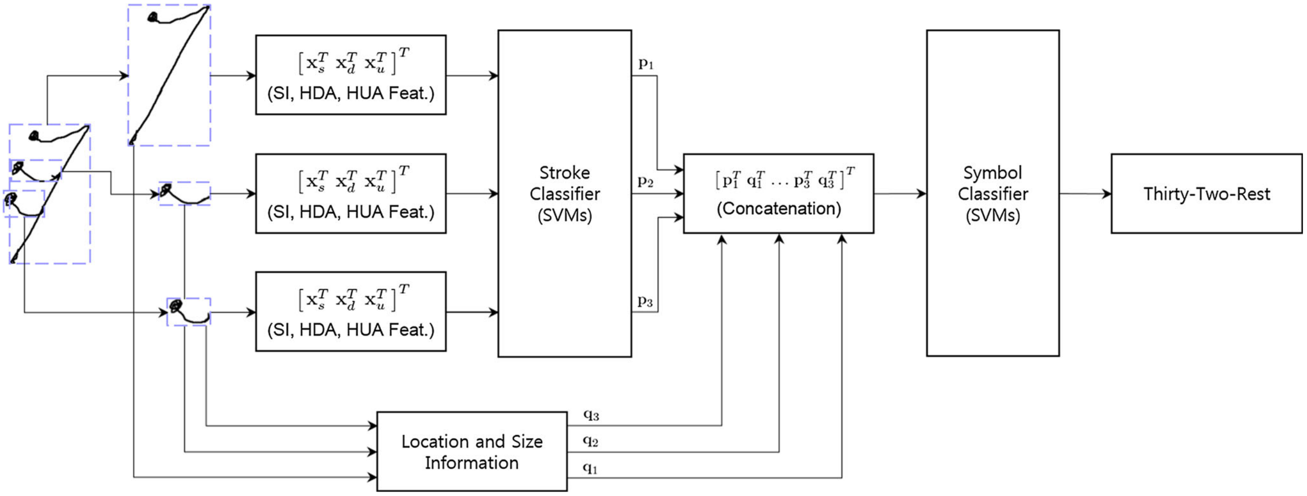

Fig. 1 Overall procedure of the proposed online music symbol recognition system

time information, conventional techniques such as dynamic time warpping (DTW) [3] and hidden Markov models [4] was applied to music symbol recognition in [5,6], and [7]. In some studies such as [8], the offline methods were applied to images of music symbols right after they have been drawn. Meanwhile, Miyao and Maruyama proposed a novel pen-based music symbol recognition system in [9]. Motivated from the fact that music symbols can be divided into common basic strokes, the system consists of two parts, one of which is to classify each input handwritten stroke as one of the basic strokes and the other is to combine the classification results into music symbols. There are two major advantages in this approach. First of all, the segmentation, which is one of the most challenging problems in music symbol recognition, is not needed for stroke classification. Secondly, various music symbols can be recognized by classifying relatively small number of basic strokes by a combination rule. It was experimentally shown that their system can recognize handwritten music symbols with a good performance. However, the system requires the users to follow some rules in writing specific strokes, but most people including composers do not draw the strokes according to the rules.

In this paper, we propose an online music symbol recognition method as shown in Fig. 1. It has a similar structure as [9], but users of the proposed method do not have to be aware of any writing rules. To achieve the goal, the proposed method utilizes three types of features: the size information (SI), the histogram of directional movement angles (HDA), and the histogram of undirected movement angles (HUA). The original size of a stroke, which was used in [9] only to recognize the stroke for a dot, is important to distinguish other strokes as well without any prior knowledge. It is also noticeable that the additional performance improvement can be obtained by assigning more weights to the SI features than to the HDA and HUA features. The HDA features contain a time-series information, which is useful in handwriting recognition and is encoded by the chain code as in [9]. Especially, the HDA features are inspired by the descriptor of the scale invariant feature transform [10] and the histogram of oriented gradients (HOG) features [11], both of which are popular in various computer vision applications. Together with the HDA features, the HUA features are also used to deal with the cases where the HDA features are not discriminant enough to classify strokes precisely. Based on the results of stroke classification using the three types of features, the proposed method recognizes each music symbol in a similar way to [9], where music symbols are recognized based on the combination of the classified strokes. However, different from [9], our symbol recognition method is performed by exploiting additional information on the sizes and locations of the classified strokes together with their combination. This enables the proposed method to recognize a set of music strokes as one of the music symbols more accurately compared to the previous methods. The proposed method is trained and verified using the HOMUS dataset [12], which was recently released and is the largest online handwritten music symbol dataset. Figure 2 shows examples of twenty four symbols in a subset of the dataset used in this study[1]. Note that the intra-variation in each class (symbol) is very large. Nonetheless, the experiments indicate that our method gives a better performance than one of the state-of-the-arts.

This paper is organized as follows. In the next section, the three types of features used in the proposed method are described. In Sect. 3, the method of music symbol recognition is mentioned. In Sect. 4, experimental results demonstrate that the proposed method is effective in recognizing handwritten music symbols. Finally, Sect. 5 concludes this paper.

[1] Originally, the dataset consists of thirty-two music symbols. However, we excluded eight symbols corresponding to time signatures.

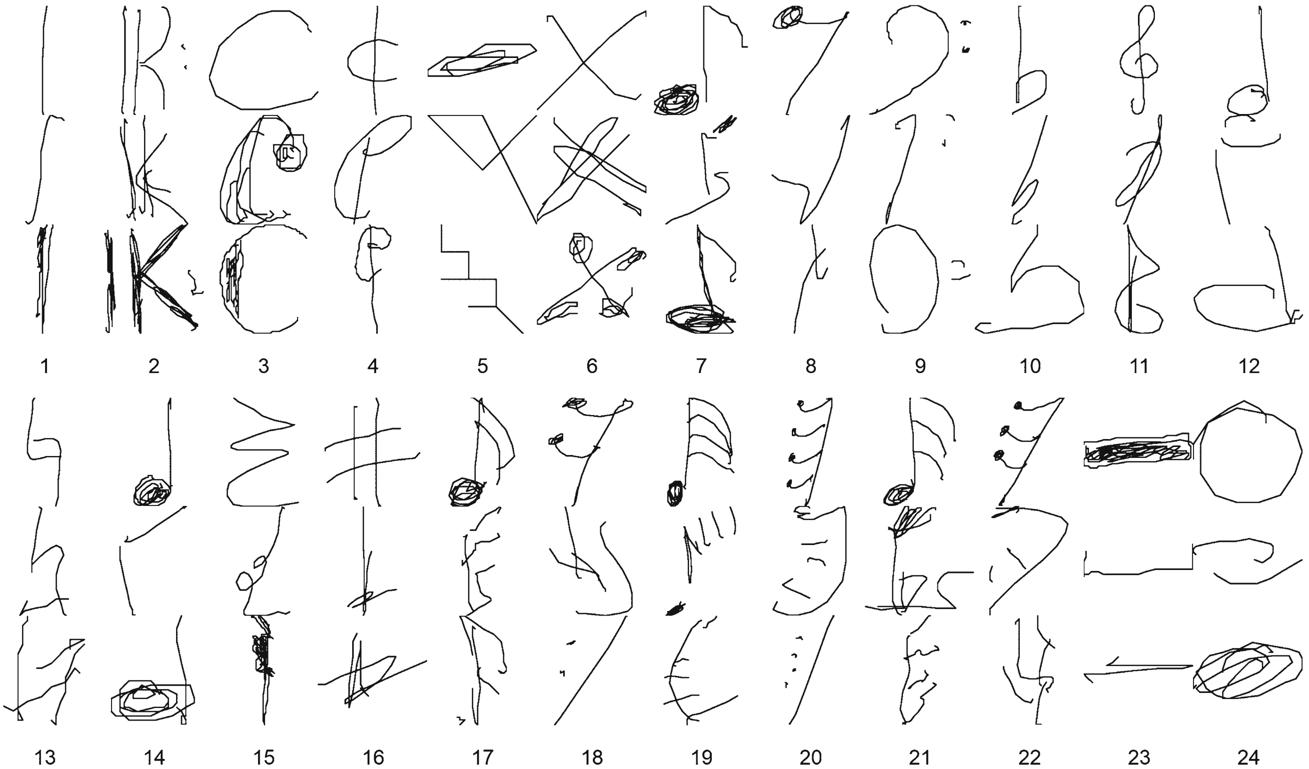

Fig. 2 Examples and the corresponding class labels of twenty four music symbols in a subset of HOMUS dataset [12] used in this study. For each music symbol, three examples are shown columnwise

2 Stroke classification

As in [9], we consider a music symbol as a set of strokes, and a stroke is defined as a sequence of two-dimensional points $\mathbf{s}_i = (x_i, y_i)$, $i = 1, \ldots, S$, which are the successive locations of a stylus pen on a device in time sequence while the pen touches the device, i.e,. the stroke is regarded to start with a pen-down and end with the pen-up. Figure 3 shows the examples of twenty-three classes of handwritten music strokes which can comprise twenty four music symbols in the HOMUS dataset as shown in Fig. 2. For the visualization, each stroke is normalized into a square region keeping the aspect ratio and consecutive two points of a stroke are connected by a line. In order to effectively recognize those strokes, we compute the following three types of features. The following subsections describe how to compute the features specifically.

2.1 SI features

A user of the system proposed in [9] has to follow a specific rule to draw a filled note head, which corresponds to BHead (black head) in Fig. 3. However, most people without the knowledge of the rule generally draw BHead like a large dot. Actually, after being normalized to the same size, the strokes corresponding to Dot and BHead have very similar shapes as

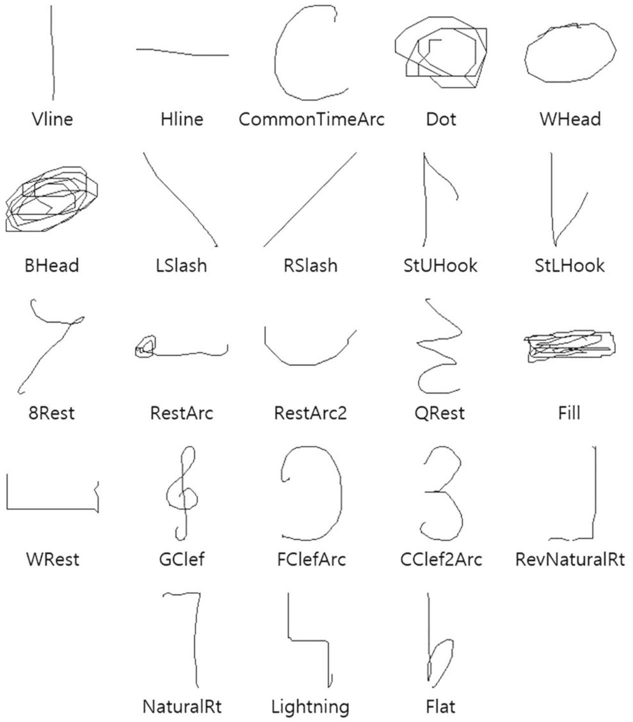

Fig. 3 Examples of twenty-three strokes which can comprise music symbols in the HOMUS dataset

shown in Fig. 3. In this case, their original sizes can play an important role in discriminating them from each other. Motivated by this observation, we generate three features related to the size of a stroke $\{\mathbf{s}_i = (x_i, y_i)\}_{i=1}^{S}$ as the following:

$$w_s = \max_i(x_i) - \min_i(x_i),$$
$$h_s = \max_i(y_i) - \min_i(y_i),$$
$$l_s = \sum_{i=1}^{S-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2},$$

where w_s and h_s are the width and the height of the box surrounding the stroke, respectively, and l_s corresponds to the length of the stroke. Under the assumption that staff lines are given, the three values are divided by the staff height, which means the vertical distance between the highest and the lowest staff lines, because most people usually write music symbols depending on the size of a staff.

2.2 HDA features

For online recognition of handwriting, the time-dependent information is so useful that it has been widely utilized in many studies [13,15,16]. Also in [9], the time-varying information was used in the framework of the eight-directional Freeman chain code [17]. The chain code is composed of a series of eight numbers which are assigned to discretized angles corresponding to a two-dimensional pen-movement. We also use the discretized angles of the directional movements, but make a histogram of the numbers assigned to the directional movement angles instead of enumerating the numbers in time sequences. More specifically, all the strokes are first normalized into a square with a fixed size of 72×72. Then, several histograms are constructed from the $d \times d$ subregions in the square instead of making one histogram from the given stroke, i.e., an eight-bin histogram of the directional movement angles is constructed in each subregion as in [10,11]. Figure 4a shows how to assign eight numbers to the directional movement angles, which is the same as the assignment of the eight-directional chain code. The directional movement $\mathbf{d}_i$ is obtained from the i-th point in a stroke as in [18]

$$\mathbf{d}_i = \begin{cases} \mathbf{s}_{i+1} - \mathbf{s}_i & \text{for i = 1,} \\ \mathbf{s}_i - \mathbf{s}_{i-1} & \text{for i = S,} \\ \mathbf{s}_{i+1} - \mathbf{s}_{i-1} & \text{otherwise,} \end{cases} \qquad (1)$$

where S is the number of 2D points in a stroke. The location of the point (x_i, y_i) determines which subregion the angle of its directional movement $\mathbf{d}_i$ belongs to. Figures 5a and 5b visualize a stroke associated with BHead and the HDA features generated from the stroke with $d = 4$. Note that the HDA feature does not depend on the number of classes. However, the chain code of an input stroke, which was used in [9], has to be matched to every reference chain code of all the classes by the dynamic programming [19]. Also, the matching should be performed once more for each class due to the ambiguity of starting point of the stroke.

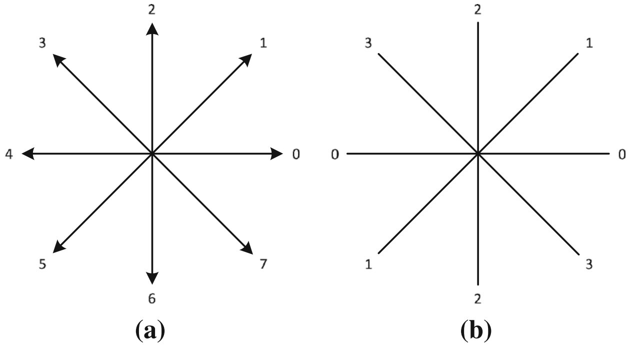

Fig. 4 **a** How to assign numbers to directional movement angles for HDA features. **b** How to assign four numbers to undirected movement angles for HUA features

2.3 HUA features

Most people write a stroke corresponding to WHead in Fig. 3 as a circle, but some people write it clockwise while the others write it counterclockwise. Using only the HDA features, the two strokes, which are differently drawn in those ways, are located far away from each other in the feature space, and this may degrade the classification performance. To prevent the performance degradation, the proposed method utilizes the HUA features as well as the HDA features. The HUA features are generated in a similar manner as the HDA features, i.e., they are computed based on the directional movements in (1), and the similar histograms of four bins are also computed from $u \times u$ subregions of a normalized stroke. The only difference between the two features is that the HDA features have the directions, whereas the HUA features do not have them as shown in Fig. 4. Actually, if d is equal to u, the HUA features can be obtained from the HDA features by summing the values of two opposite directional bins of the HDA features. Figure 5b, c show the relation between the HDA and HUA features. In this case, the HDA and HUA features may yield the same information because the HUA features can be represented as a linear combination of the HDA features. However, their parameters d and u were actually determined as different values, which will be mentioned in Sect. 4. The HUA features look similar to the directional image features, which were first presented in [13] and were also used in online handwritten music symbol recognition system presented in [9]. However, there is a significant difference between the

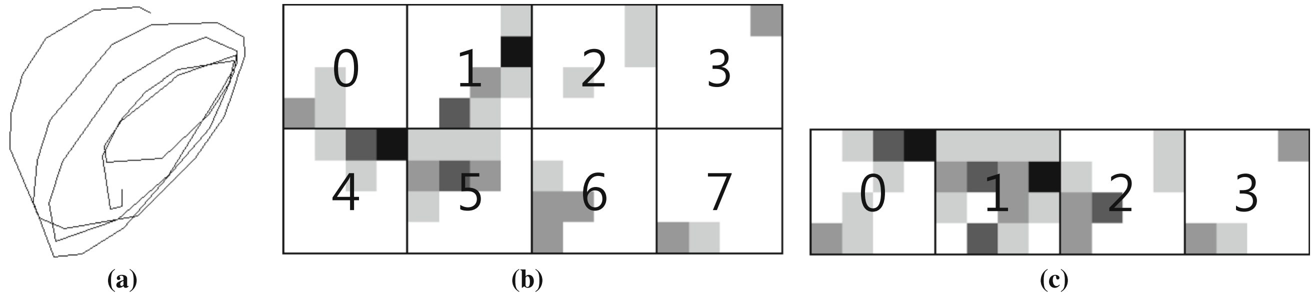

Fig. 5 A stroke of BHead and the corresponding HDA and HUA features. **a** The stroke. **b** Its HDA features with $d = 4$. **c** Its HUA features with $u = 4$. *Black* (*white*) denotes high (low) value of histogram. The numbers on the visualized features correspond to the numbers assigned to movement angles as shown in Fig. 4

proposed features and the image directional features. The HUA features are directly computed from the normalized points of a stroke while the image directional features are generated from an image of the stroke after converting the stroke to the image. Thus, it can be said that the HUA features can save the computational effort to transform input strokes to images compared to the image directional features. In the computations of the HDA and HUA features, the subregions can be overlapped as in the block normalization of the HOG features [11]. However, we do not consider the overlaps because the features computed without the overlapping provide enough performance, which will also be shown in Sect. 4.

2.4 Classification

To classify handwritten music strokes, we propose to use the three types of the above mentioned features by concatenating them, i.e., a stroke to be recognized is represented by the following feature vector:

$$\mathbf{x} = \left[\, \mathbf{x}_s^T \ \mathbf{x}_d^T \ \mathbf{x}_u^T \,\right]^T , \tag{2}$$

where $\mathbf{x}_s$, $\mathbf{x}_d$ and $\mathbf{x}_u$ are the feature vectors corresponding to the SI, HDA, and HUA features, respectively. If both of d and u are set to 4 as shown in Fig. 5, the dimensionality of $\mathbf{x}$ becomes 195 because $\mathbf{x}_s \in \Re^3$, $\mathbf{x}_d \in \Re^{d\times d\times 8}$, and $\mathbf{x}_u \in \Re^{u\times u\times 4}$. In order to classify the feature vectors constructed as in (2), we employ the kernel SVM [14], which is the nonlinear extension of the linear SVM and finds the hyperplane satisfying the maximum margin criterion in a higher dimensional feature space induced by a nonlinear mapping. By means of the kernel trick, the hyperplane defined in the feature space can be obtained without explicit knowledge of the nonlinear mapping. Let us consider a training set $\{(\mathbf{x}_i, c_i)\}_{i=1}^N$ where $\mathbf{x}_i$ is the feature vector of the i-th training sample, which can be computed as (2), and $c_i \in \{1, -1\}$ is the class label of the sample. When using the kernel SVM, the hyperplane can be represented as

$$\gamma = \sum_{i=1}^{N} \alpha_i c_i k(\mathbf{x}_i, \mathbf{x}) + \beta, \tag{3}$$

where $\{\alpha_i\}_{i=1}^N$ and β are obtained from the SVM learning, and an arbitrary sample $\mathbf{x}$ can be classified depending on the sign of (3) once the learning is completed. In the above equation, $k(\mathbf{x}_1, \mathbf{x}_2)$ is the kernel between $\mathbf{x}_1$ and $\mathbf{x}_2$, and it is used as the value of the inner product between the two vectors in the feature space. Although various functions can be employed for the kernel, we use the Gaussian kernel defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\sigma^2}\right),$$

where $||\mathbf{x}||$ is the Euclidean norm of a vector $\mathbf{x}$, and σ is the kernel parameter. Since σ deeply affects the performance of SVM classifier, determining its value is important in the design of a SVM classifier using the Gaussian kernel. If the training set is inseparable, SVM has another tuning parameter C to adjust the trade-off between the margin and the penalty that enables the training samples to be misclassified. We find the appropriate values of the parameters (σ, C) by cross-validation, which will be described in Sect. 4.

While SVM was developed for binary classification problems, the strokes to be classified in this study belong to one of $M = 23$ classes. In this work, we adopt the one-against-the rest (OAR) scheme to design a classifier for the multi-class classification problem using multiple SVMs because the scheme is the most popular. In the scheme, the number of SVMs required for M-class classification problems is M, and each SVM is trained to recognize whether a test sample belongs to a class or not. When an input stroke is given, the output values in (3) are computed from the M SVMs and the stroke is classified as one of the M classes providing the maximum output value.

In summary, an input stroke $\{\mathbf{s}_i\}_{i=1}^{S}$ is classified as one of M classes as follows. It is first encoded as a feature vector $\mathbf{x}$ in (2). Then, the feature vector is applied to M SVMs and the output of this stroke classification is as the following:

$$\begin{aligned} \mathbf{p} &= \left[\, p_1 \ p_2 \ \dots \ p_M \,\right]^T , \\ p_i &= \frac{\exp(\gamma_i)}{\sum_{i=1}^{M} \exp(\gamma_i)} \end{aligned} \tag{4}$$

where γ_i is the output of the i-th SVM in (3) and p_i corresponds to the probability that the feature vector $\mathbf{x}$ belongs to the i-th class.

3 Symbol recognition based on stroke classification

In this part, a music symbol is recognized based on the above stroke classifications as in [9], where only the combination of the classification results was utilized in symbol recognition step. Unlike [9], together with their combination, we additionally consider the spatial relation among the classified strokes. In order to encode the spatial relation, we use the location and size of strokes along with the stroke classification output $\mathbf{p}$. Let us consider an input music symbol consisting of K strokes. For example, the four symbols in Fig. 6 consist of one, two, three, and four strokes, respectively, where each stroke is denoted as a dashed rectangle in the figure. We propose to define a feature vector for symbol recognition as

$$\begin{aligned} \mathbf{z} &= \left[\, \mathbf{p}_1^T \ \mathbf{q}_1^T \ \dots \ \mathbf{p}_K^T \ \mathbf{q}_K^T \,\right]^T , \\ \mathbf{q}_i &= \left[\, \overline{x}_i \ \overline{y}_i \ \overline{w}_i \ \overline{h}_i \,\right]^T , \end{aligned} \tag{5}$$

where $\mathbf{p}_i$ is the classification result of the i-th stroke defined in (4) and $\mathbf{q}_i$ contains the location and size information of the i-th stroke. Specifically, $\overline{x}_i$ and $\overline{y}_i$ are the center location of the stroke and $\overline{w}_i$ and $\overline{h}_i$ are its width and height. To consider the relative locations and sizes of strokes belonging to a symbol instead of their absolute values, the elements of $\mathbf{q}_i$ are normalized to be between zero and one based on the symbol region where they are included. Using this construction, a music symbols of $K \geq 2$ strokes is represented as a feature vector $\mathbf{z}$ in a $K \times (4 + 23)$-dimensional feature space. Two samples corresponding to the same symbol can have different dimensions if they have different values of K. When $K = 1$, the feature vector is defined as

$$\left[\, \mathbf{p}_1^T \ \frac{\overline{w}_1}{\max(\overline{w}_1, \overline{h}_1)} \ \frac{\overline{h}_1}{\max(\overline{w}_1, \overline{h}_1)} \,\right]^T$$

because its location ($\overline{x}_1$ and $\overline{y}_1$) is not informative to discriminate it. In this approach, it is necessary to train a classifier to recognize music symbols for each value of K. Fortunately, the number of the classifiers for symbol recognition is not large. Actually, over 99% of 12,000 symbols in a subset of the HOMUS data set are composed of $K \leq 6$ strokes as shown in Fig. 7. For each value of K, multiple SVMs are trained as in Sect. 2.4 and the details will be described in Sect. 4.

When computing the feature vectors in (5), the concatenation order of strokes can be important because two samples of a symbol with the same $K \geq 2$ can be far away from each other in the feature space if the order is not consistent.

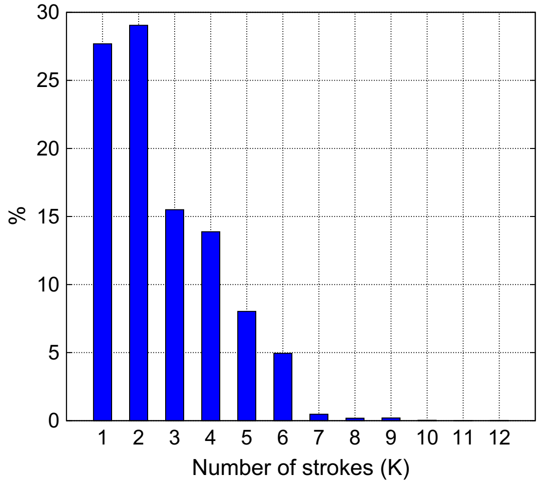

Fig. 7 Distribution of numbers of strokes in each symbol in a subset of HOMUS dataset

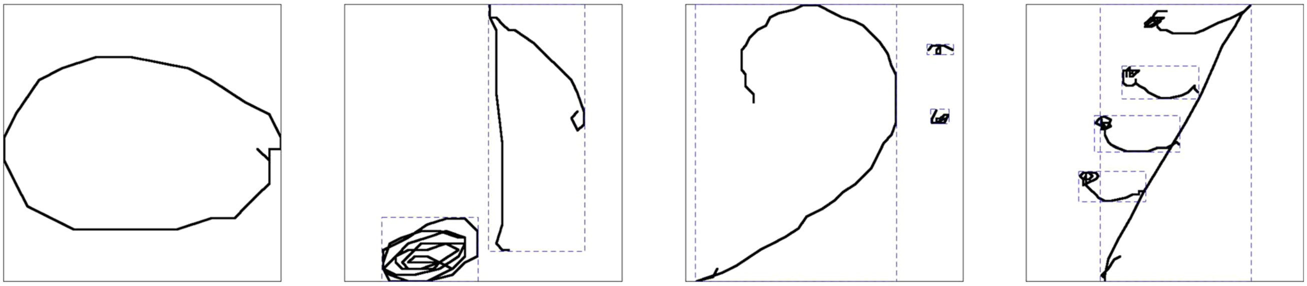

Fig. 6 Examples of music symbols consisting of one, two, three, and four strokes

We consider two concatenation orderings in constructing a feature vector in (5). The first is to make the feature vector by concatenating $\left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$ in written order using the advance of the online technique. The other is to concatenate $\left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$ in the lexicographical ordering based on $\left[\overline{x}_i, \overline{y}_i\right]$. In this ordering, $\left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$ is ahead of $\left[\mathbf{p}_j^T, \mathbf{q}_j^T\right]^T$ if $\overline{y}_i < \overline{y}_j$ or $\overline{x}_i \leq \overline{x}_j$ when $\overline{y}_i = \overline{y}_j$. The performances of the two orderings will be compared experimentally in the next section.

4 Experiments

For the purpose of training and evaluating the proposed method, we used a subset of the HOMUS dataset [12] and the SNU dataset [2] which will be described later in detail. All the experiments in this paper were performed using MATLAB running on a 3.6 GHz Intel Core i7 PC.

The HOMUS dataset contains twenty four music symbols as shown in Fig. 2. It does not include eight time signatures in the original dataset. In the subset, every note symbol except whole note contains 800 samples and the other symbols contains 400 samples so that it consists of 12,000 samples in total. In order to train the proposed method, the labels of strokes are necessary. However, although the study on music stroke clustering using the dataset was recently presented in [20], the labels are not served in the original dataset. We analyzed 31,768 strokes in all the samples and chose twenty-three basic strokes in Fig. 3.[3] In this process, we tried to define as small number of the basic strokes as possible keeping the similarity between any pair of strokes to be as small as possible. As a result, each basic stroke contains somewhat large variations. With those basic strokes, we labeled all the strokes as one of the twenty four classes, which is summarized in Table 1. ‘None’ in the table contains the strokes that can not be categorized into any of the twenty-three basic strokes. When labeling the samples of Dot symbol, some of the samples were regarded as ‘None’ because their sizes were as large as BHead strokes, but this can be compensated in the symbol recognition step, which was shown in [9].

The proposed method has two types of parameters to be tuned. The first one is related to SVM, i.e., (σ, C), and the other is related to the feature vector $\mathbf{x}$ for stroke classification, e.g., d and u. In the following subsections, we describe how to tune the parameters.

Table 1 The numbers of strokes comprising music symbols in a subset of HOMUS dataset

Stroke	# of strokes	Stroke	# of strokes
None	4281	RestArc	554
VLine	5377	RestArc2	890
HLine	1222	QRest	152
CommonTimeArc	810	Fill	324
Dot	1888	WRest	89
WHead	1053	GClef	388
BHead	2904	FClefArc	913
LSlash	3662	CClef2Arc	161
RSlash	3719	RevNaturalRt	35
StUHook	362	NaturalRt	262
StLHook	1211	Lightning	98
8Rest	1151	Flat	262

4.1 Tuning parameters

The samples of the HOMUS dataset was collected from 100 different musicians. Using all the samples in the dataset, 100-fold cross-validation was performed to evaluate various techniques in [12]. However, different from [12], we tuned and evaluated the proposed method by 10-fold cross-validation for efficiency. Specifically, we divided all the symbol samples into ten sets such that the symbol samples collected from each musician belong to one of ten sets and each set consists of the samples gathered from ten musicians. Thus, it can be said that our experiments correspond to the user-independent experiment in [12]. Using the ten sets, cross-validations were conducted to determine each of the parameters in the proposed method. In the process of cross-validations, training samples were normalized to have zero-mean and unit variance, and each test sample was also normalized using the means and standard deviations of the training samples.

- HDA and HUA parameters: a stroke defined as $\{\mathbf{s}_i = (x_i, y_i)\}_{i=1}^{S}$ is needed to be converted to a feature vector in (2) for classification. In this conversion, two parameters d and u, which are the parameters of HDA and HUA features, respectively, should be predetermined. We varied the parameters d and u such that $d, u \in \{2, 4, 6, 8\}$, and set them as the values providing the best performance by the 10-fold cross-validations. In more details, after d or u was set to one of the values in the above set, 10-fold cross-validations were performed to find the best pair of (σ, C) for each of twenty-three SVMs from the following candidate sets:

$$\sigma \in \{10, 50, 100, 150, 200\},$$
$$C \in \{5, 10, 50, 100, 200\}.$$

[2] The SNU dataset is available at http://mipal.snu.ac.kr/index.php/Repository.

[3] The labels of the stokes are also available at http://mipal.snu.ac.kr/index.php/Repository.

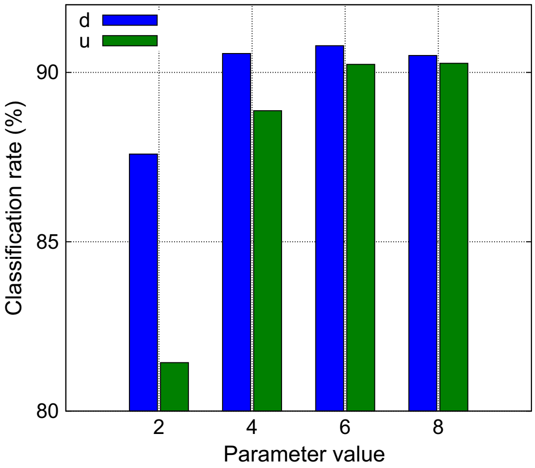

Fig. 8 Performances of stroke classification for different values of d and u

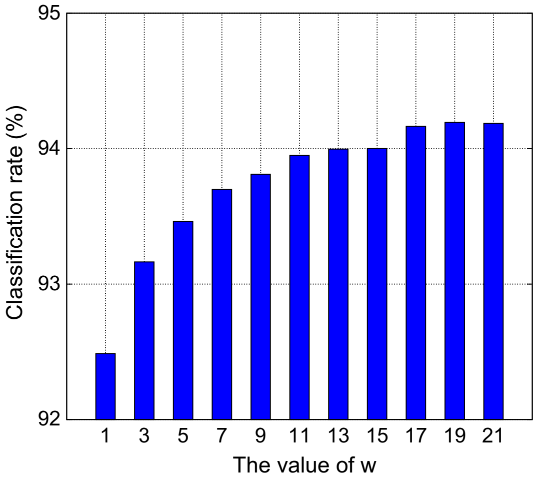

Fig. 9 Performances of stroke classification for different values of w

After determining the best pairs of (σ, C) for the twenty-three SVMs, the value for d or u was evaluated by an OAR-based classifier using the SVMs which were trained with the best pairs of (σ, C). Figure 8 shows the performance of the values for d and u. Since $d = 6$ and $u = 8$ gave the highest performances of 90.79% and 90.27%, respectively, they are fixed as the values in the following experiments. Note that d and u were set to different values, so that the HDA and HUA features can provide complementary information to discriminate music strokes.

- Weights of SI features: the above choice of HDA and HUA parameters yields $\mathbf{x}_d \in \Re^{288}$ and $\mathbf{x}_u \in \Re^{256}$, which makes a dimensionality imbalance between $(\mathbf{x}_d, \mathbf{x}_u)$ and $\mathbf{x}_s \in \Re^3$. In order to compensate the imbalance, we propose to assign a higher weight w to $\mathbf{x}_s$ after the normalization of zero-mean and unit variance. As a result of this compensation, the feature vector for stroke classification in (2) is changed to

$$\mathbf{x} = \left[\, w\mathbf{x}_s^T \;\mathbf{x}_d^T \;\mathbf{x}_u^T \,\right]^T . \tag{6}$$

The value of w was also found by the 10-fold cross-validation like the tuning of d and u. We conducted the 10-fold cross-validations by increasing the value of w by two starting from one. The validation was repeated until the classification rate is less than the previous one. Figure 9 shows the performances of the 10-fold cross-validations obtained as varying the value of w. As shown in the figure, the performance increased as the value increased and the best performance was obtained when it was equal to 19. Thus, we set the value of w to 19 in the next experiments.

4.2 Stroke classification

Figure 10 shows the results of stroke classification for each stroke by using different combinations of the features. Note that the classification rate of Dot stroke was dramatically improved by using the SI feature and assigning suitable weights to them. Although the performance improvements were not as large as Dot stroke, we could observe similar trends of the improvement in CommonTimeArc, WRest and GClef strokes. In total, the single HDA and HUA features gave 90.79% and 90.27%, respectively, and we could obtain a slightly higher classification rate 91.02% by combining the two types of features. Utilizing the SI features together with HDA and HUA features provided 92.49%, which is about 1.5% higher classification rate compared to the one without the SI features. Furthermore, the assignment of appropriate weights to the SI features additionally improved the classification performance by 1.7% (from 92.49 to 94.19%). This indicates that the SI features can be complementary to the HDA and HUA features in classifying music strokes for online music symbol recognition.

4.3 Music symbol recognition

As in the previous experiments, our method to recognize music symbols was trained and evaluated by 10-fold cross-validation using the ten subsets mentioned in Sect. 4.1. Once a pair of training and test sets was given, all of the symbol samples in the training set were divided according to the number of strokes (K) in the samples. For each value of K, an SVM-based classifier was trained using the feature vectors computed from the symbol samples as in (5) in the written order (Time) and the lexicographical order (Lexi). The parameters of each SVM in the classifier were determined in a similar manner described in Sect. 4.1. In test phase, all the strokes in a symbol sample were first classified, and its feature vector was then computed in the same way. The sample was finally recognized as one of the twenty four music symbols by the trained SVM-based classifier. For the purpose of

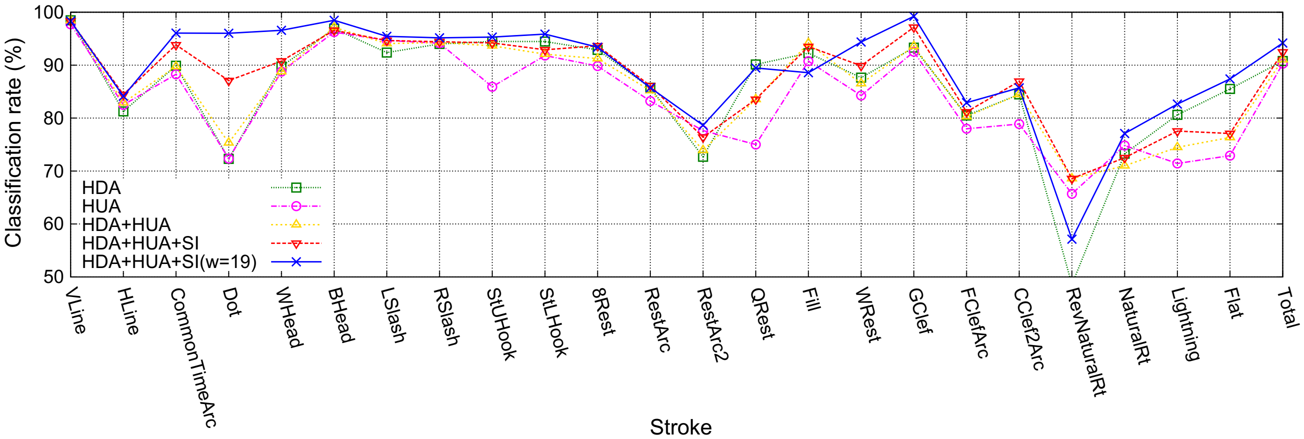

Fig. 10 Performances of stroke classification for different combinations of the features. Best viewed in color

comparison, we also implemented DTW [3], which showed the best performance in [12]. The performance of DTW was also measured by the same 10-fold cross-validation. For each symbol sample in test set, we computed its dissimilarities from all of the symbol samples in training set by DTW, and it was recognized by the one nearest neighbor (1NN) and the three nearest neighbor (3NN) classifiers.

Furthermore, we collected handwritten music symbols besides HOMUS dataset to additionally evaluate the proposed method. The music symbols were gathered from twenty-three subjects including students majoring music composition at a university, which was named as the SNU dataset. When collecting the dataset, we used the same device under a similar setting, which was reported in [12]. The symbol samples in SNU dataset were used only for testing in this experiment, i.e., we chose the symbol samples corresponding to the twenty four symbols as shown in Fig. 2. Table 2 summarizes the numbers of the selected samples. The stroke and symbol classifiers of the proposed method were trained using the HOMUS dataset only. The parameter values of the features for stroke classification and those in SVMs were kept unchanged from the ones determined from HOMUS dataset.

Table 2 The numbers of symbols in SNU dataset

Symbol class	# of symbols	Symbol class	# of symbols
1	263	13	80
2	0	14	108
3	0	15	90
4	0	16	111
5	81	17	140
6	0	18	72
7	142	19	0
8	71	20	0
9	90	21	0
10	88	22	0
11	147	23	54
12	108	24	71

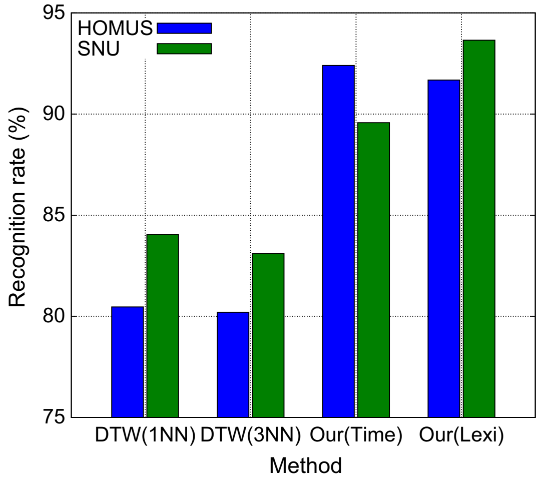

Fig. 11 Music symbol recognition performances

Figure 11 shows the performances of the proposed method and DTW for two datasets. Note that the maximum recognition rates of the proposed method are 92.40% (Time) and 93.65% (Lexi) for HOMUS and SNU dataset, respectively, which are 11.94% and 9.62% higher than those of DTW. This means that the proposed method is effective enough to be used in online handwritten music symbol recognition. We could observe in Fig. 11 that one of the two concatenation orders does not give better performances than the other in the both datasets. Figure 12 visualizes the confusion matrices obtained by the proposed method with the written and lexicographical orders. The largest difference between the two confusion matrices is the recognition performance of class 16 (Sharp). It was only about 25% with the written order, but it increased up to about 97% by changing the order.

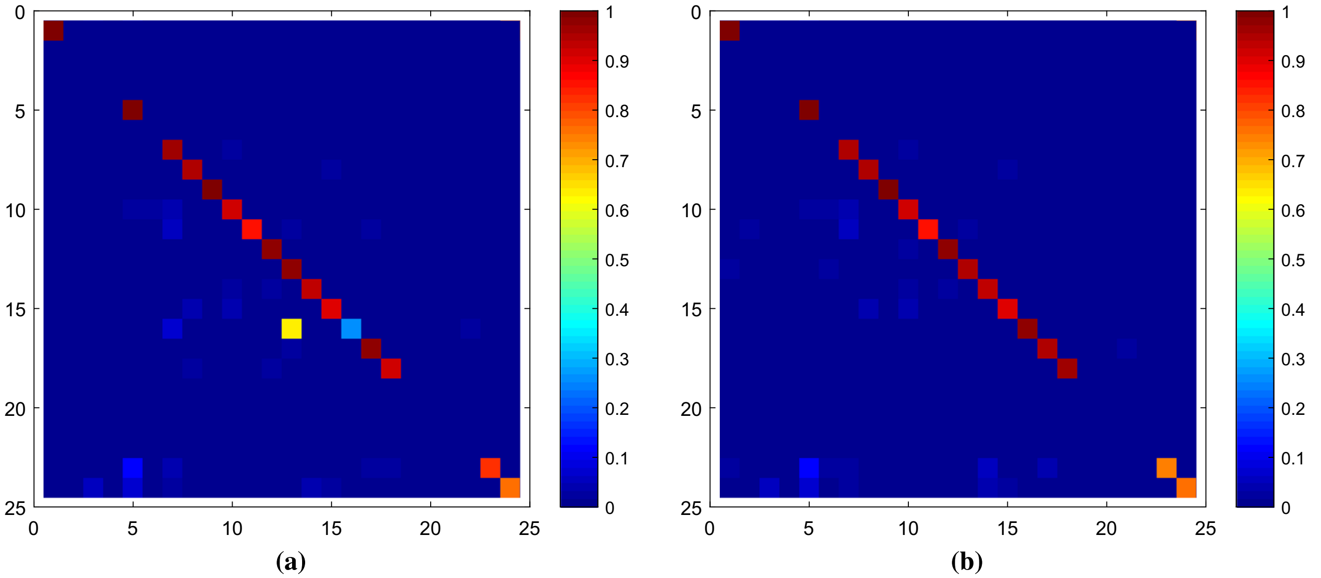

Fig. 12 Confusion matrices obtained from concatenating in **a** written order and **b** the lexicographical ordering

Table 3 Average computation time (in millisecond)

Proposed method				DTW
Feature extraction	Stroke classification	Symbol classification	Total	
0.24	174.69	1.52	465.16	394.66

This inconsistency is attributed to the fact that the strokes in the symbols of Sharp in HOMUS and SNU datasets were written in different sequences. Thus, we expect that the approaches for online music symbol recognition, which recognize music symbols by classifying its strokes and combining the classification results, can be enhanced by developing an effective algorithm to integrate classified strokes into a music symbol regardless of the concatenation order.

In addition to accuracy, we also evaluated computation time of the proposed method in comparison with DTW. We measured the computation time of testing procedure for both the methods using HOMUS dataset. Table 3 shows the average computation time. It can be seen that the proposed method takes 18% more time than DTW to recognize a music symbol on average. However, the increase of computation time is acceptable enough considering the accuracy improvement of over 10% (from 80.46% to 92.40%). We further analyzed the computation time of our method that can be considered to be composed of three steps, i.e., the feature extraction, the stroke classification, and the symbol classification. The average computation time for each step is also shown in Table 3. Note that the sum of the times for the three steps is not equal to the total time. It is because a music symbol consists of several music strokes, but the computation time for the stroke classification in the table was measured for a single stroke. We can see from the table that the stroke classification step consumes most computation time among the three steps. In general, the computational complexity of SVM classification increases with the dimensionality of input data and the number of support vectors determined by SVM training. The dimensionality of **x** in (6) is 547, and the dimensionalities of **z** in (5) for most symbols are less than 100. Also, it is known that the number of support vectors selected in SVM training tends to increase as the number of training samples increase. Since a symbol consists of several strokes, the number of training samples for the stroke classifiers is larger than the number of training samples for the symbol classifiers. From these results, we can expect that the computation time to recognize an arbitrary input music symbol by our system can decrease by employing the directed acyclic graph (DAG) [21] scheme of SVMs for the classifiers in the proposed method instead of the OAR scheme.

5 Conclusion

We proposed a music symbol recognition method in this paper. The proposed method is similar to the previous method presented in [9]. However, different from the method, our method does not require users to follow any writing rule. To achieve the goal, we used three types of features, which are the SI, the HDA, and the HUA features. Compared to the previous method, those features are simple and efficient to generate because they are directly computed from raw input data, which are 2D points, without the conversion to image. We achieved additional improvement by assigning more weights to SI features, which are much lower dimensional than HDA and HUA features. It may be possible to apply the HDA and HUA features to other online handwriting recognition problems. In addition, different from the previous method, we utilized the relative sizes of the strokes and their spatial relation in addition to the combination of the strokes when combining the classified strokes into a music symbol. The proposed method was evaluated using the HOMUS and the SNU datasets. It provided about 10% higher recognition rates than DTW, which is one of the state-of-the-arts, for both datasets. This is one of the main contributions of this study. Those experimental results demonstrate that the proposed method is effective in recognizing music symbols written without any specific rule.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2016R1A1A1A05005442) and by the Information Technology Research Center support program through the Institute for Information & communications Technology Promotion (IITP-2016-R0992-16-1012), both of which are funded by the Ministry of Science, ICT and Future Planning, Korea.

References

1. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A., Guedes, C., Cardosom, J.: Optical music recognition: state-of-the-art and open issues. Int. J. Multimed. Inf. Retr. **1**, 179–190 (2012)
2. Rebelo, A., Capela, G., Cardoso, J.: Optical recognition of music symbols. Int. J. Doc. Anal. Recognit. **13**, 19–32 (2010)
3. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**, 43–49 (1978)
4. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
5. Calvo-Zaragoza, J., Oncina, J., Iñesta, J.: Recognition of online handwritten music symbols. In: Proceedings of 6th International Workshop on Machine Learning and Music (MML13) (2013)
6. Lee, K.C., Phon-Amnuaisuk, S., Ting, C.Y.: Handwritten Music Notation Recognition Using HMMa non-gestural approach. In: Proceedings of 2010 International Conference on Information Retrieval and Knowledge Management (CAMP), pp. 255–259 (2010)
7. Hu, L., Zanibbin, R.: hmm-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature. In: Proceedings of 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 457–462 (2011)
8. George, S.: Online pen-based recognition of music notation with artificial neural networks. Comput. Music J. **27**, 70–79 (2003)
9. Miyao, H., Maruyama, M.: An online handwritten music symbol recognition system. Int. J. Doc. Anal. Recognit. **9**, 49–58 (2007)
10. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
11. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
12. Calvo-Zaragoza, J., Oncina, J.: Recognition of pen-based music notation: the HOMUS dataset. In: Proceedings of 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 3038-3043 (2014)
13. Okamoto, M., Yamamoto, K.: On-line handwriting character recognition using direction-change features that consider imaginary strokes. Pattern Recognit. **32**, 1115–1128 (1999)
14. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
15. Plamondon, R., Srihari, S.N.: On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 63–84 (2000)
16. Bahlmann, C.: Directional features in online handwriting recognition. Pattern Recognit. **39**, 115–125 (2006)
17. Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput. **EC–10**, 260–268 (1961)
18. Bai, Z.-L., Huo, Q.: a study on the use of 8-directional features for online handwritten chinese character recognition. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition, vol. 1, pp. 262–266 (2005)
19. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recognit. **26**, 1797–1812 (1993)
20. Calvo-Zaragoze, J., Oncina, J.: Clustering of strokes from pen-based music notation: an experimental study. In: Proceedings of Seventh Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), pp. 633–640 (2015)
21. Platt, J.-C., Cristianini, N., Shawe-taylor, J.: Large margin DAGs for multiclass classification, Advances in Neural Information Processing Systems, pp. 547–553 (2000)