IJDAR
DOI 10.1007/s10032-010-0114-8

FULL PAPER

Rotation invariant hand-drawn symbol recognition based on a dynamic time warping model

Alicia Fornés · Josep Lladós · Gemma Sánchez · Dimosthenis Karatzas

Received: 3 October 2009 / Revised: 8 February 2010 / Accepted: 18 February 2010

Abstract One of the major difficulties of handwriting symbol recognition is the high variability among symbols because of the different writer styles. In this paper, we introduce a robust approach for describing and recognizing hand-drawn symbols tolerant to these writer style differences. This method, which is invariant to scale and rotation, is based on the dynamic time warping (DTW) algorithm. The symbols are described by vector sequences, a variation of the DTW distance is used for computing the matching distance, and K-Nearest Neighbor is used to classify them. Our approach has been evaluated in two benchmarking scenarios consisting of hand-drawn symbols. Compared with state-of-the-art methods for symbol recognition, our method shows higher tolerance to the irregular deformations induced by hand-drawn strokes.

Keywords Document analysis · Graphics recognition · Symbol recognition · Handwriting recognition · Sequence alignment

A. Fornés (✉) · J. Lladós · G. Sánchez · D. Karatzas
Computer Vision Center, Department of Computer Science,
Universitat Autònoma de Barcelona, Edifici O,
08193 Bellaterra, Spain
e-mail: afornes@cvc.uab.es

J. Lladós
e-mail: josep@cvc.uab.es

G. Sánchez
e-mail: gemma@cvc.uab.es

D. Karatzas
e-mail: dimos@cvc.uab.es

1 Introduction

Symbol recognition is one of the main topics of Graphics Recognition, which has been an intensive research work in the last decades, covering technical symbol recognition [37], handwritten symbol recognition [2], symbol indexing and spotting [34], or even the recognition of degraded symbols (e.g. [45,47,48]).

Graphical languages are expressive and synthetic tools for communicating ideas in some domains. A graphical language consists of an alphabet of symbols and the rules defining the valid combinations among them. Thanks to the recognition of the alphabet of symbols of these graphical languages, combined with domain-dependent knowledge, the whole document has a meaning, allowing its automatic processing.

Hand-drawn symbol recognition is a particular case of handwriting recognition, which is one of the most significant topics within the field of document image analysis and recognition (DIAR). Over the last years, relevant research achievements have been attained. Simultaneously, commercial products have become available. The progress has been noticeable in applications like bank check processing, postal sorting, historical document transcription or online recognition in calligraphic interfaces. A parallel use has also been explored in writer identification for forensic sciences and writer verification in signatures. Handwriting recognition is a difficult problem due to the variability among scripts and writer styles, or even between different time periods. Due to that, commercial applications are usually constrained to controlled domains that make use of contextual or grammatical models and dictionaries. The type of source data (handwritten separate characters vs cursive script) is also an important constraint. Focusing on cursive script recognition, the recognition approaches can roughly be classified into *analytical* or *holistic* methods. Analytical methods perform

Published online: 10 March 2010

Fig. 1 High variability of hand-drawn musical clefs: **a** Treble, **b** Bass, **c** Alto

Fig. 2 Distorted shapes: **a b** Distortion on junctions, **c** Gaps, **d** Overlapping, **e** Missing parts

a segmentation preprocess that divides the word image in sequences of smaller units which are therefore classified in terms of associated features and lexical information. Holistic methods, which recognize words as a whole, usually describe the word image as a unidimensional signal consisting of a sequence of image features at each column. This allows to use techniques sometimes inspired by the speech recognition domain such as sequence alignment by dynamic programming [16] or Hidden Markov Models [32].

In this paper, we present a novel rotation invariant symbol recognition method without restricting its applicability. We choose to focus here on the case of hand-drawn graphical symbols of non-textual alphabets as a representative problem. This refers to symbols that compound diagrammatic notations in graphical documents like musical scores, architectural drawings, electronic and engineering diagrams, and flow charts. (see [21] for a review). In addition to the high writer style differences (see Fig. 1) and the inherent distortion of hand-drawn strokes (see Fig. 2), the recognition of graphical symbols has two added difficulties regarding to handwritten text recognition. First, graphical symbols are bidimensional shapes appearing in bidimensional layouts, so 1D models should be adapted to rotation, scale, and position invariance. Second, unlike text, graphical symbols can not benefit from the use of contextual and grammatical models.

To cope with the problem of hand-drawn symbol recognition under the conditions stated in the above paragraph, in this work we propose a method inspired by the holistic approaches for unconstrained handwritten word recognition but extended to bidimensional shapes appearing in bidimensional layouts. Our main contribution is an approach to model and classify hand-drawn symbols. The proposed method is robust against the elastic deformations typically found in handwriting and invariant to rotation and scale. The method proposed is based on the dynamic time warping (DTW) algorithm [16] for signals (one-dimensional data), and it has been extended to graphical symbols (two-dimensional data). Among the two major families of methods for handwriting recognition, namely sequence alignment (e.g. DTW) and Hidden Markov Models (HMMs), our work is based on the former. The DTW algorithm has been successfully used for finding the best match between two time series in a noisy and complex domain. It has been already used in handwritten text recognition [33], coping with the elastic deformations and distortions in the writing style. For that reason, we maintain that the DTW algorithm can be adapted for the recognition of hand-drawn symbols. In comparison with HMMs, the DTW approaches are more suitable for coping with the problem of hand-drawn symbol recognition when there is a small number of instances for each symbol (which is the case of some hand-drawn graphical databases), not being enough for a successful training process. In addition, the adaptation of DTW to a rotation invariant system is easier than the adaptation of HMM, because HMM requires to train a model for each possible orientation, with the consequently increment of its time complexity.

To solve the problem of rotational invariance, classical and effective methods exist in the literature on OCR or Symbol Recognition. Methods like projections in different orientations or zoning using concentric ring masks are well known. We have taken into account these ideas and extended them to a novel DTW-based algorithm. The steps of the method proposed are the following: First, column sequences of feature vectors from different orientations of the two input shapes to be compared must be computed. The features comprise the upper and the lower profiles and the number of pixels per region. Once we have the features for all the considered orientations, the DTW algorithm computes the matching cost between every orientation of the two symbols and decides in which orientation these two symbols match with the lowest cost.

The rest of the paper is organized as follows. Section 2 corresponds to the state of the art of hand-drawn symbol recognition methods. In Sect. 3, the fundamentals of the dynamic time warping (DTW) algorithm are presented. Afterward, our DTW-based method for the recognition and classification of graphical symbols is fully described, demonstrating its invariance to rotation and scale. In Sect. 4, the experimental results are presented. Finally, concluding remarks are exposed in Sect. 5.

2 State of the art of hand-drawn symbol recognition methods

Hand-drawn symbol recognition has been one of the most intensive research fields of graphical symbol recognition [21]. It is close to handwritten character recognition,

especially for logographic languages such as chinese characters [5,20]. In fact, some of these approaches use variants of DTW (see [39] for a survey).

In the handwriting domain, symbol recognition methods require symbol descriptors with three important properties: first, they should guarantee intra-class compactness and inter-class separability; second, they should be rotation and scale invariant; and third, they should cope with elastic deformations and distortions caused by the high variability in handwritting style.

Traditionally, symbol descriptors, as a particular case of shape descriptors, can be classified into statistical and structural approaches. The first ones represent the image as a n-dimensional feature vector, whereas the second ones usually represent the image as a set of geometric primitives and relationships among them. Statistical approaches tend to use pixels as the primitives to extract features from. The curvature scale space (CSS), Zernike moments, Generic Fourier Descriptor, Radial Angular Transform, and Shape Context descriptors are examples of these statistical approaches. The CSS [26] descriptor only takes into account the symbol silhouette and can only be used for closed curves, but it is tolerant to rotation. On the contrary, Shape Context [2] can work with non-closed curves and has good performance in hand-drawn symbols, because it is tolerant to deformations, but it requires point-to-point alignment of the symbols to be compared before their alignment. The generic Fourier descriptor (GFD) [46] applies a 2D Fourier Transform to the polar representation of the image and is rotation and scale invariant. The angular radial transform (ART) [14] decomposes the shape in an orthogonal basis, taking use of a radial and angular function. It has good performance for general shapes, and it is robust to noise. Zernike moments [13] are widely used for hand-drawn symbols, as well as online systems [10], because they preserve properties of the shape and are invariant to rotation, scale, and deformations. There are also several statistical approaches for online symbol recognition, which can also use on-line information such as speed or pressure. Although they are usually more focused in human interfaces, a few works are briefly mentioned next: In [30], a method applied to logic diagrams is proposed, which uses geometric features and template matching. In the method proposed in [43], the symbol is represented as a sequence of coordinates, and matching is based on curvature distance. Miyao and Maruyama [25] present a hand-drawn music symbol recognition system, consisting of the combination of two classifiers: the first one uses chain codes for representing the strokes, while string-edit distance is used for matching; the second classifier is used for complex strokes, dividing strokes into regions, and computing the directional feature for each region. Golubitsky and Watt [9] propose the recognition of multi-stroke symbols using truncated Legendre–Sobolev expansions of the coordinate functions for creating the feature vectors and classifying using support vector machines.

In structural approaches, straight lines and arcs are usually the basic primitives. Strings, graphs, or trees represent the relations between these primitives. The similarity measure is therefore performed by string, tree, or graph matching. A few examples of structural approaches are briefly described next: the attributed graph grammars [3] can cope with partially occluded symbols, while Spectral models [19] and Region Adjacency Graphs [22] are well-suited to describe symbols in hand-drawn documents, showing good performance in front of distortions typically found in these documents. Deformable models [40] are invariant to distortions and rotation, but the basic primitives are lines, thus not being suitable for symbols with arcs and curves. Hidden Markov Models are also widely used in offline [27] and online symbol recognition methods [44]. Basically the structure of the symbol is described by the sequence of states that generate the image, and the recognition consists in finding the sequence of states with the highest probability. Concerning structural approaches for online symbol recognition, Fonseca et al. [6] propose a method for recognizing architectural symbols, using fuzzy logic and geometric features; Peng et al. [31] propose a constrained partial permutation algorithm which uses binary and ternary topological spatial relationships for the recognition of symbols; and Mas et al. [24] describe a complete system for recognizing architectural drawings, representing the data as trees and proposing adjacency grammars with distortions measures for adapting them to sketches.

Mathematical symbol recognition requires a mixed strategy, because it requires text recognition and graphics (symbols) recognition. It is a very active research field (see [4] for a survey), which also includes several online systems: Shi et al. [36] propose a symbol decoding and graph generation algorithm; and a full mathematical expression recognizer system is defined in [8], which involves symbol recognition (using both online and offline features) and structural analysis of multi-stroke characters using context-free grammars.

3 A DTW-based approach for graphical symbol recognition

Since the approach proposed in this paper is based on the DTW algorithm, we will start this section with a short introduction before detailing our approach. The DTW algorithm was first introduced by Kruskal and Liberman [16] for putting series into correspondence. This technique was first used in the context of speech recognition, a domain in which the time series are notoriously complex and noisy. The method was used for coping with noise and variations in speech speed. Beside speech recognition, this technique has been

widely used in many other applications: chemical engineering, gesture recognition, signatures, robotics, bioinformatics, music, shape retrieval, or Data Mining [1,11,12,29]. DTW has been also applied to the handwritten text recognition field, being used in both offline [15,17,33] and online approaches [28,42].

The basic dynamic time warping algorithm achieves good results when working with one-dimensional data and with handwritten words in documents. Concerning the hand-drawn symbol domain, the method must be adapted to cope with the variations in writing style and rotation. In the first part of this section, the fundamentals of DTW are presented. Afterward, the architecture for our DTW-based system is fully described, and its benefits for hand-drawn symbol recognition are presented. Compare to the classical DTW, the proposed method introduces two main changes: first, different features are used and second, the computation of the DTW distance has been modified, combining information at certain orientations of the symbol.

3.1 DTW for one-dimensional signals

The DTW algorithm [16] is used for comparing signals by matching two one-dimensional vectors. It is a much more robust distance measure for time series than Euclidean distance, allowing similar samples to match even if they are out of phase in the time axis (see Fig. 3). DTW can distort (or warp) the time axis, compressing it at some places and expanding it at others, finding the best matching between two samples.

Let us define the DTW distance of two time series $C = x_1 \dots x_M$ and $Q = y_1 \dots y_N$ as DTWCost(C, Q) (see Fig. 4a). For this purpose, a matrix $D(i, j)$ (where $i = 1 \dots M$, $j = 1 \dots N$) of distances is computed using dynamic programming:

$$D(i, j) = \min \left\{ \begin{array}{c} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{array} \right\} + d2(x_i, y_j) \quad (1)$$

$$d2(x_i, y_j) = x_i - y_j \quad (2)$$

Performing backtracking along the minimum cost index pairs (i, j) starting from (M, N) yields the warping path (Fig. 4b). Finally, the matching cost is normalized by the length Z of this warping path, otherwise longest time series should have a higher matching cost than shorter ones:

$$\text{DTW Cost}(C, Q) = D(M, N)/Z \quad (3)$$

The creation of this path is important, because it determines which points match (Fig. 4c) and are to be used to calculate the distance between the time series. In addition, DTW is able to handle samples of unequal length, allowing the comparison without resampling.

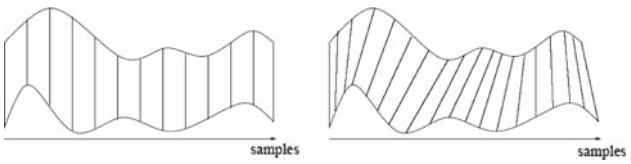

Fig. 3 Normal and DTW alignment, extracted from [33]

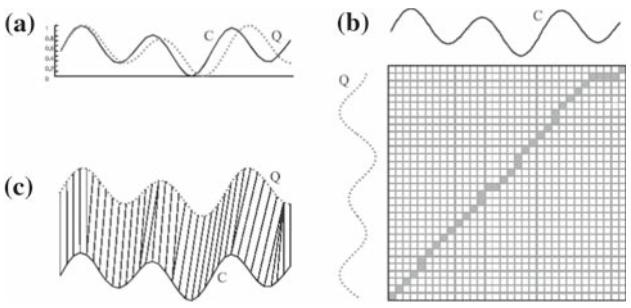

Fig. 4 An example of DTW alignment (extracted from [12]) **a** Samples C and Q, **b** The matrix D with the optimal warping path in gray color, **c** The resulting alignment

3.2 DTW for two-dimensional shapes

In case of bidimensional data, the DTW computation must be adapted. Some work has been done in the adaptation of DTW to 2 dimensions [18,38], but these approaches are of a very high time complexity, reaching $O(N^{4N})$ and $O(N^3 9^N)$ respectively. For this reason, some research work has been focused on the reduction of the 2D problem. Generally, the reduction of dimensionality can be performed when 2D data can be encoded by 1D signals, such as shapes described by their external contours (silhouettes). Specifically, for handwritten text methods, the 2D representation is typically reduced to 1D based on the assumption that text follows a given text line [33]. In these cases, the time complexity of the 2D-DTW computation is significantly reduced.

3.3 Extraction of features

The choice of features that better represent shapes is a key decision of the application of the DTW algorithm. In this work, we have been inspired by features representing series with a view to reduce the dimensionality. Let us first describe the approaches which have inspired our proposed representation.

In the handwritten text recognition system described by Rath and Manmatha [33], the following four features are computed for every column of a word image: the number of foreground pixels in every column; the upper profile (the distance of the upper pixel in the column to the upper boundary of the word's bounding box); the lower profile (the distance of the lower pixel in the column to the lower boundary

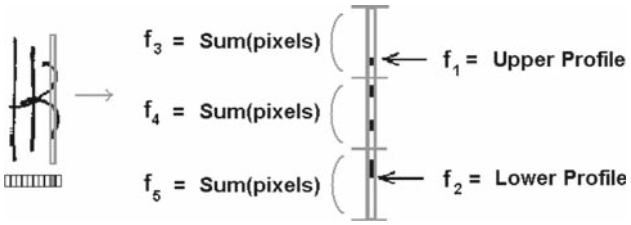

Fig. 5 Example of features extracted from every column of the image, with $S = 5$: $f_1 =$ upper profile, $f_2 =$ lower profile, $f_3 \ldots f_5 =$ sum of pixels of the image of the three regions defined

of the word's bounding box); and the number of transitions from background to foreground and viceversa. In this way, two word images A and B can be easily compared using DTW. If $f_k(a_i)$ corresponds to the k-th feature of the column i of the image A, and $f_k(b_j)$ corresponds to the k-th feature of the column j of the image B, the matching distance DTW Cost(A, B) is calculated using the same equations (Eq. 1, 3) as in Kruskal's method, but instead of Eq. 2, the computation of $d2$ will be the sum of the squares of the differences between individual features:

$$d2(x_i, y_j) = \sum_{k=1}^{4} (f_k(a_i) - f_k(b_j))^2 \quad (4)$$

Another typical set of column features in the literature is the one proposed by Marti and Bunke [23] for handwritten word recognition. The following nine features are obtained per column: the number of foreground pixels, the center of gravity, the second moment order, the lower and upper profiles, the differences between the lower and upper values with respect to the previous column, the number of gaps, and the number of pixels between the upper line and baseline of the word. Finally, the features described by Vinciarelli et al. [41] are also very common in the literature, consisting in a sliding window which moves from left to right. In this case, instead of the single column features, the window comprises several columns. After adjusting the size of the window to the area which contains pixels, it is divided into a 4x4 cell grid, and the number of pixels in every cell is used as a feature. Then, the 4x4 features are concatenated to a 16-dimensional feature vector.

Inspired by the above approaches, we propose a feature set for symbol recognition. In this field, it is important to obtain some information about the external shape (profiles), but also about the internal shape (distribution of pixels inside the silhouette). In fact, in other recognition fields (e.g. chinese character recognition), it has been demonstrated that the external profiles (e.g. the peripheral features) are not efficient enough for the recognition of certain characters [5]. For this reason, in addition to the upper and the lower profiles, our method divides every column in several regions, counting the number of foreground pixels per region (it can be seen as a column zoning). First, the image is normalized in terms of its size, and the following features are computed for every column of the image:

- $f_1 =$ upper profile.
- $f_2 =$ lower profile.
- $f_3 \ldots f_S =$ number of foreground pixels in each region.

When computing the upper and lower profiles, a morphological closing operation over the image is performed, so that few little gaps in the writing will not affect the final profile. Finally, all the features are normalized ($0 \leq f_k \leq 1$, k = 1…S), and the features corresponding to the sum of pixels (f_3,…,f_S) are smoothed over the symbol's columns using a gaussian filter for a better matching. Notice that due to the high variability in the writing style, the number of transitions per column (from background to foreground and viceversa) can confuse the system, thus, they are not used as features.

Figure 5 shows an example of the features extracted for the marked column of a music symbol: the pixels of the column are used for extracting the upper and the lower profiles. Then, the column is divided in three equal regions (in this example, S = 5), and for every region the number of pixels is counted.

The reader should notice that the features $f_3, \ldots, f_S$ provide an adequate information about the distribution of the pixels inside the shape. The number of regions is a parameter that can be set up to reflect the complexity of the symbols in the database. These measures will help to classify correctly shapes that have the same external contour but differences in their interior. Moreover, it will not get confused when comparing axially symmetrical symbols. In Fig. 6b, one can see two similar images in terms of silhouette (both are squares), but very different inside (a cross or a circle). Notice that the upper/lower profiles and the whole sum of pixels per column are very similar (Fig. 6a), whereas the functions of the sum of the three regions (see Fig. 6c) are very different, being able to discriminate the symbols.

3.4 Computation of the DTW distance

Due to the fact that the slant and the orientation of graphical symbols are frequently different between each other (see Fig. 7), symbols can not be directly and easily compared between them.

To cope with rotation invariance and hand-drawn distortion, we define a DTW-based distance in terms of different projections, covering the full range of possible orientations of the symbol.

Let us introduce the notation that will be used in this section:

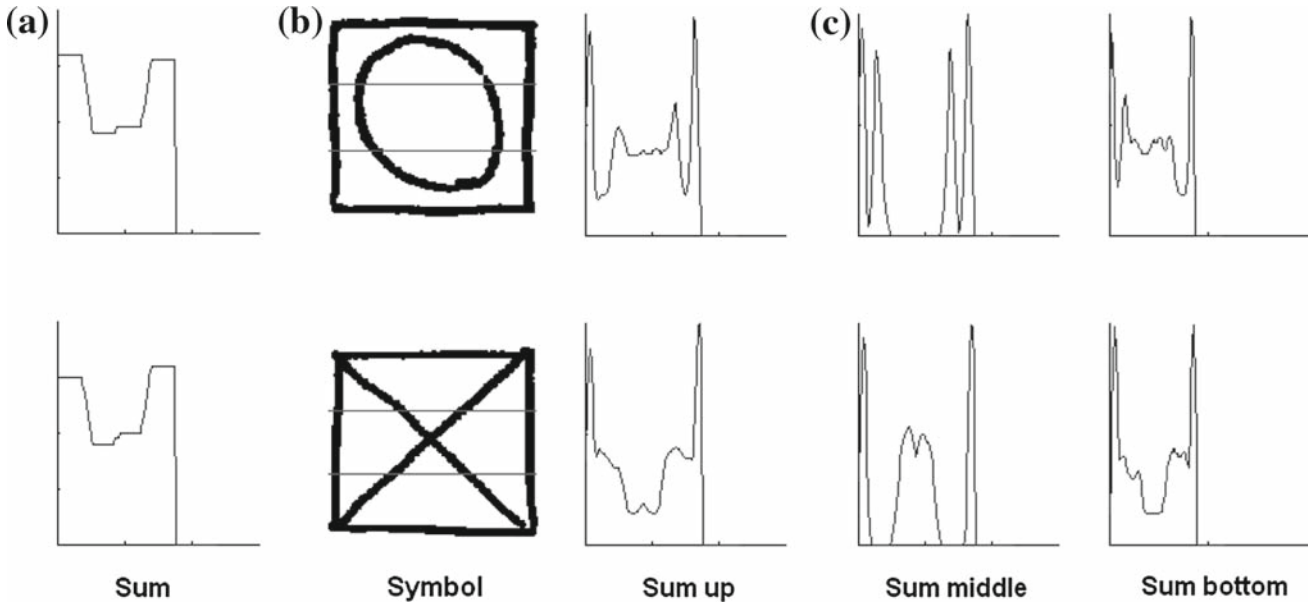

Fig. 6 Two architectural symbols with similar external contour (*squares*) but with differences inside the contours (*circle* and *cross*). The first row corresponds to the features for the square with a circle, and the second row corresponds to the features for the square with a cross. **a** Functions of the sum of pixels per column. **b** Symbols. The gray horizontal lines divide the image in three regions: upper, lower, and middle **c** Functions corresponding to the sum of pixels for the upper, middle, and bottom region. Notice that the functions in **a** are similar, whereas functions in **c** are very different

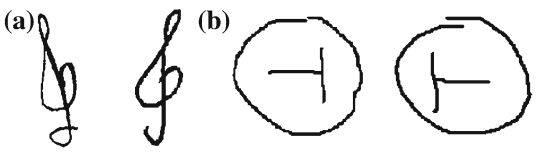

Fig. 7 **a** Clefs: two treble clefs with different slants, **b** Two identical architectural symbols but in different orientations

- A_α: Symbol A oriented at α degrees.
- B_β: Symbol B oriented at β degrees.
- $\mathtt{a}_{\alpha_i}$: Column i of the symbol A oriented at α degrees.
- $\mathtt{b}_{\beta_j}$: Column j of the symbol B oriented at β degrees.
- $D_{\alpha,\beta}(i, j)$: Matrix which contains the cost of matching the first i columns of A_α and the first j columns of B_β.
- $MC(\alpha, \beta)$: Matrix which contains at the position (α, β) the matching cost between A_α and B_β.
- $G(\alpha, \beta)$: Matrix which contains at the position (α, β) the sum of $MC(\alpha, \beta)$ and $MC(\alpha + 90, \beta + 90)$.

There are three steps in the procedure: the extraction of features at different orientations; the computation of the matching distance between all the possible combinations of orientations between the two symbols; and the computation of the final matching distance. In the first step, the two symbols A and B are oriented in certain angles (see Fig. 8a),

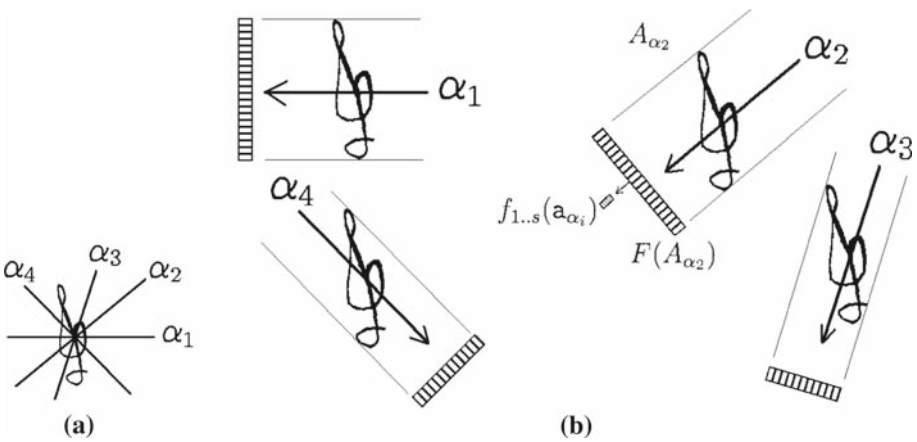

Fig. 8 Example of feature extraction. **a** Some of the orientations used for extracting the features of every symbol. **b** Feature vectors extracted from every orientation $(\alpha_1, \ldots \alpha_4)$

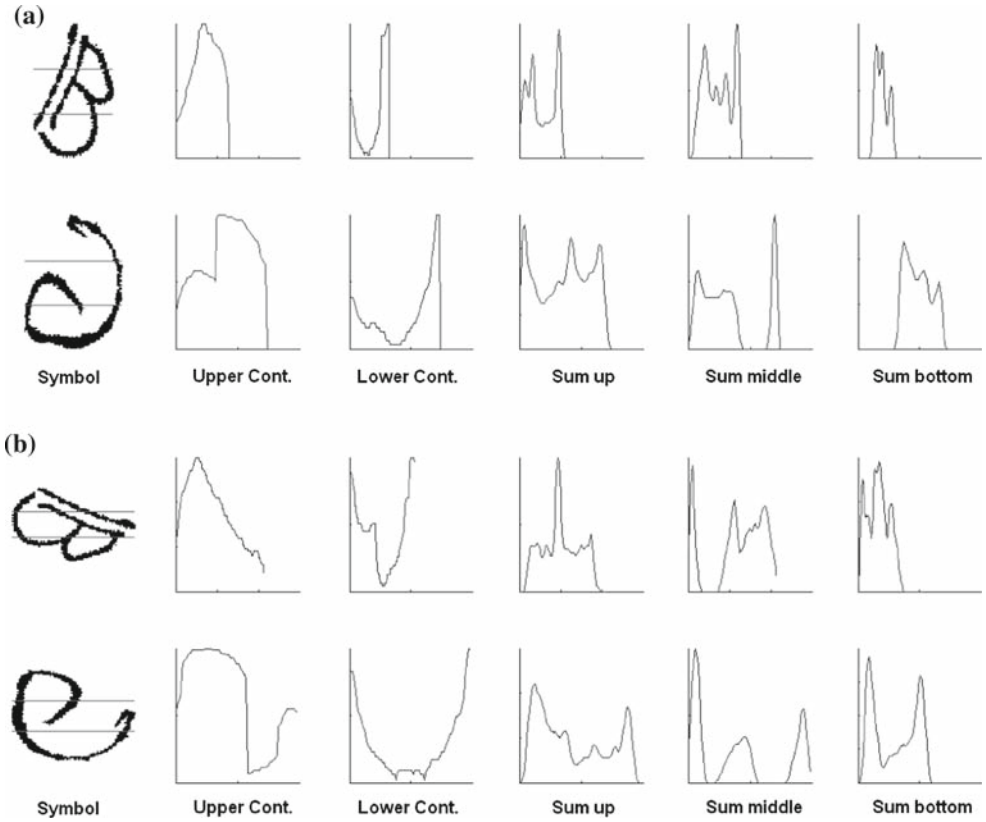

Fig. 9 Feature vectors of two different music symbols: **a** The first symbol is an alto clef with a orientation of α degrees, the second one is a bass clef with a orientation of β degrees. **b** The same alto clef with a orientation of $\alpha + 90°$ and the bass clef with a orientation of $\beta + 90°$. Here, the functions of the two symbols are very different

covering the range from 0 to 180°. For each orientation, the column sequence of feature vectors (see Fig. 8b) defined in the previous section is obtained. In the second step, the DTW distance is computed for every combination of orientations of the two symbols. Thus, every orientation of the symbol A is compared to every orientation of the symbol B. It should be observed that it is necessary to obtain the features from every orientation of the two symbols, because we do not know a priori which orientation will give the highest discriminatory power. Finally, the third step consists in determining the final matching cost, and the two angle orientations in which the two symbols match with the lowest cost. In fact, we can not trust in only one matching when working with 2D data, because false matchings could appear if only one direction is used (see Fig. 9). For this reason, we also take into account the perpendicular alignment in respect to the orientation we are considering. As a summary, we can define the final matching cost $\mathrm{DTWCost}_{A,B}$ of the symbol A and B as the minimum of the results of summing $MC(\alpha, \beta) + MC(\alpha + 90, \beta + 90)$ for each possible α, β angles.

Let us define as $A_\alpha = (\mathtt{a}_{\alpha_1}, \mathtt{a}_{\alpha_2}, \ldots, \mathtt{a}_{\alpha_M})$ the symbol A oriented at α degrees, and $B_\beta = (\mathtt{b}_{\beta_1}, \mathtt{b}_{\beta_2}, \ldots, \mathtt{b}_{\beta_N})$ the symbol B oriented at β degrees. First, the column sequences of feature vectors $F(A_\alpha)$ and $F(B_\beta)$ are computed as it has been explained in the above section (the upper/lower profile and the sum of pixels per region):

$$F(A_\alpha) = \begin{pmatrix} f_1(\mathtt{a}_{\alpha_1}) & f_1(\mathtt{a}_{\alpha_2}) & \ldots & f_1(\mathtt{a}_{\alpha_M}) \\ f_2(\mathtt{a}_{\alpha_1}) & f_2(\mathtt{a}_{\alpha_2}) & \ldots & f_2(\mathtt{a}_{\alpha_M}) \\ \ldots & \ldots & \ldots & \ldots \\ f_s(\mathtt{a}_{\alpha_1}) & f_s(\mathtt{a}_{\alpha_2}) & \ldots & f_s(\mathtt{a}_{\alpha_M}) \end{pmatrix} \tag{5}$$

$$F(B_\beta) = \begin{pmatrix} f_1(\mathtt{b}_{\beta_1}) & f_1(\mathtt{b}_{\beta_2}) & \ldots & f_1(\mathtt{b}_{\beta_N}) \\ f_2(\mathtt{b}_{\beta_1}) & f_2(\mathtt{b}_{\beta_2}) & \ldots & f_2(\mathtt{b}_{\beta_N}) \\ \ldots & \ldots & \ldots & \ldots \\ f_s(\mathtt{b}_{\beta_1}) & f_s(\mathtt{b}_{\beta_2}) & \ldots & f_s(\mathtt{b}_{\beta_N}) \end{pmatrix} \tag{6}$$

Notice that the length of every column sequence of feature vector depends on the number of columns (the width) of the projection and varies from one orientation to another.

Once the column sequences of feature vectors are computed, the matching cost $MC(A_\alpha, B_\beta)$ between them must be calculated. First, the matrix D will be filled in with the

classical DTW method:

$$D_{\alpha,\beta}(i,j) = \min \left\{ \begin{array}{l} D_{\alpha,\beta}(i,j-1) \\ D_{\alpha,\beta}(i-1,j) \\ D_{\alpha,\beta}(i-1,j-1) \end{array} \right\} + d2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) \tag{7}$$

The way of computing the distance *d2* must take into account that both the upper/lower profile features and the set of sum of pixels features have to be weighted equally in the calculation. The goal is to avoid a reduced effect of the upper/lower profile in the computation of *d2* whenever the feature number *S* is very high (which means a high number of regions for the zoning) For this reason, the two parts are weighted by 0.5 as following:

$$d2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) = 0.5 \cdot P_1(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) + 0.5 \cdot P_2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) \tag{8}$$

$$P_1(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) = \left(\sum_{k=1}^{2} (f_k(\mathtt{a}_{\alpha_i}) - f_k(\mathtt{b}_{\beta_j}) \right)^2 \tag{9}$$

$$P_2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) = \left(\sum_{k=3}^{s} (f_k(\mathtt{a}_{\alpha_i}) - f_k(\mathtt{b}_{\beta_j}) \right)^2 \tag{10}$$

Then, the matching cost of A_α and B_β is normalized by the length Z of the warping path (obtained performing backtracking on $D_{\alpha,\beta}$), and this value is stored in the corresponding cell of the matrix MC:

$$MC(\alpha,\beta) = D_{\alpha,\beta}(M,N)/Z \tag{11}$$

This process must be repeated for all the orientations $\alpha = 1 \ldots 180$ and $\beta = 1 \ldots 180$ (the step is decided ad-hoc), filling all the cells in the matrix MC. Thus, every cell of the matrix $MC(\alpha,\beta)$ will contain the matching cost between the two symbols, the first one with an orientation angle of α degrees, and the second one with an orientation angle of β degrees. This means that if the two symbols are oriented in W different angles, the DTW distance is computed W^2 times.

The next step is the computation of the final matching cost. It must be noticed that defining the final matching cost as the minimum of the DTW distances computed is not a good solution. For example, two symbols, which belong to different classes, could reach the minimum matching cost if they are oriented in some specific α and β angles, but they could have very high matching costs in other orientation angles. One way to avoid this problem is to look at the perpendicular alignment in respect to the orientation we are examining. Another option could be to take into account the matching cost of all the alignments, but it has been experimentally shown that it does not increase the discriminatory power, whereas the time complexity is increased. As an example of the problem of using only one matching, Fig. 9 shows the feature vectors of two different music symbols: in Fig. 9a, one can see that despite the two symbols being extremely different, only the upper contour and the middle sum are adequately different functions in the DTW sense, whereas in Fig. 9b all the five functions of the first symbol are very different from the ones of the second symbol. For this reason, we should claim that two symbols are correctly matched in α and β orientation angles ($\alpha \in [0 \ldots 360]$, $\beta \in [0 \ldots 360]$), only if they have a low matching cost in α and β angles but also a low matching cost in the corresponding perpendicular alignment ($\alpha + 90$ and $\beta + 90°$). For this step, let us define as G the matrix which stores in position (α,β) the cell $MC(\alpha,\beta)$ plus its corresponding perpendicular angle:

$$G(\alpha,\beta) = MC(\alpha,\beta) + MC(\alpha+90, \beta+90) \tag{12}$$

Thus, the matching cost $\mathrm{DTWCost}_{A,B}$ of the symbols A and B will be defined as the minimum value of the matrix G, where the angles θ and λ correspond to the orientation angles in which the two symbols are matched:

$$\mathrm{DTW\ Cost}_{A,B} = \min(G) \tag{13}$$

Table 1 shows the pseudo-code of the algorithm.

Finally, it must be noted that with the proposed descriptor and matching strategy, we obtain a symbol descriptor and classifier methodology which is rotation invariant and robust against typical elastic deformations present in hand-drawn symbols. Concerning the complexity of the algorithm, if w corresponds to the number of angles in which every symbol is oriented, and N is the number of columns of the widest symbol image, then the complexity is $O(W^2N^2)$, because the DTW matching distance with order $O(N^2)$ is computed W^2 times. This complexity cost is remarkably lower than $O(N^{4N})$ and $O(N^3 9^N)$ of existing 2D-DTW approaches [18,38].

Table 1 DTW-based algorithm

Given two symbols A and B:
1. Obtain $F(A_\alpha)$ for every orientation $\alpha = 0$ .. 180
2. Obtain $F(B_\beta)$ for every orientation $\beta = 0$ .. 180
3. Compute the matching cost matrix MC:
For each angle $\alpha = 0$ .. 180,
For each angle $\beta = 0$ .. 180,
Compute $MC(\alpha,\beta)$
End_For
End_For
4. Add the matching cost of every angle+90 degrees:
For each angle $\alpha = 0$ .. 180,
For each angle $\beta = 0$ .. 180,
$G(\alpha,\beta) = MC(\alpha,\beta) + MC(\alpha+90,\beta+90)$
End_For
End_For
5. Find the minimum:
$DTWCost_{A,B} = min(G)$

4 Results

For the evaluation of our approach, we first describe the databases, metrics, comparisons, and experiments performed.

4.1 Benchmarking data

Two benchmarking databases of hand-drawn symbols have been used, namely music symbols from musical scores and architectural symbols from a sketching interface in a CAD framework. These two databases have been chosen for different purposes. First, with the clefs database, we plan to analyze the robustness of the proposed approach against deformations. The data set is extracted from modern and old music scores, and it is used because of the high variability of the symbols, with important elastic deformations produced by the different writer styles. With the architectural database, we evaluate the scalability with an increasing number of classes. The architectural data set contains an important number of different classes with different appearance, while the inter-class variability is comparably lower.

4.2 Benchmarking methods

Some benchmarking methods are chosen to compare our proposed features and our full DTW approach. The goal is to analyze the performance of our method but also the suitability of the set of features we propose.

Zernike moments [13], Generic Fourier Descriptor [46], Angular Radial Transform [14], and a DTW-cyclic method are used for comparing our DTW approach. Zernike moments, GFD and ART are classical shape description methods in the literature. They have been used in symbol recognition methods, because they are robust to deformations and invariant to scale and rotation. In our experiments, GFD has a radial frequency with value 4, and angular frequency with value 9; ART has a radial order with value 2, and angular order with value 11; and 7 moments are used for Zernike.

We have also implemented a variation of our own method, named cyclic DTW. The idea is to see how the performance changes when using an algorithm with a lower computational cost. It consists of taking the center of mass of the symbol, and for every orientation (from 0 to 180, with a step of 10°) we only take into account the column that corresponds to the center of mass of the shape, and for this "centroid column", the features used in our approach are computed (the upper and lower profiles, the sum of pixels per region). Thus, only one feature vector describes the symbol in every orientation. Then, a DTW-cyclic approach (similar to a string matching cyclic) is used to match the matrices of the two symbols.

Concerning feature comparison, Marti's [23] and Rath's [33] features are compared against our features. In these experiments, our DTW approach has been applied using these features from the literature, which have been described in Sect. 3. Thus, we compare the proposed features against the ones defined by Rath and Marti to establish the suitability of our features.

Referring the method proposed in this paper, we use the upper and lower profiles, and the sum of pixels of 3, 4, or 5 regions. The features are extracted from every orientation, from 0 to 180°, also with a step of 10°.

4.3 Classification

For the classification of the symbols, one representative per class is usually chosen. Thus, every input symbol of the database is compared to these n representatives, and only n comparisons are computed for classifying every input symbol. Notice that with this approach, no training process is required, saving an important computational cost. The K-nearest neighbor (in our case, 1-NN) is used as the distance for the classification. The minimum distance will define the class where the input symbol belongs to.

4.4 Music clefs data set

The data set of music clefs was obtained from a collection of modern and old musical scores (19th century) of the Archive of the Seminar of Barcelona. This database contains a total of 2,128 samples between the three different types of clefs from 24 different authors. These images have been obtained from original documents using a semi-supervised segmentation approach [7]. The main difficulty of this database is the lack of a clear class separability because of the variation of the writer styles and the lack of a standard notation. The high variability of clefs' appearance from different authors can be observed in the segmented clefs of Fig. 1.

Under this scenario, the selection of the representative for each class is not easy. The printed clefs that are shown in Fig. 10a–c are not similar enough to the hand-drawn ones. For this reason, we have chosen some hand-drawn

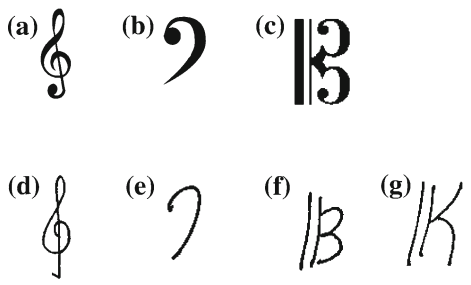

Fig. 10 Printed Clefs and Selected representative clefs: **a** Printed Treble clef, **b** Printed Bass clef, **c** Printed Alto clef, **d** Treble representative clef, **e** Bass representative clef, **f** and **g** Two Alto representative clefs

Table 2 Classification of clefs (%): recognition rate (RR.), recall and fall-out of these three music classes using four models

Method	Zernike moment	ART	GFD	DTW-cyclic	DTW-approach (5 zones)
RR. trebble	87.7	69.3	98.7	27.1	96.2
RR. bass	63.8	82.3	68.1	91.4	96.5
RR. alto	75.7	97.1	69.7	78.0	97.1
Overall rec. rate	**75.7**	**82.9**	**78.8**	**65.5**	**96.6**
Overall precision	**80.3**	**87.5**	**83.2**	**68.2**	**96.9**
Overall fall-out	**11.9**	**9**	**10.2**	**19.6**	**1.8**

representative clefs: one treble clef (Fig. 10d), one bass clef (Fig. 10e), and two alto clefs (Fig. 10f,g) because of the high variability in alto clefs. The selected representatives correspond to the set median symbol.

Given a database consisting of a set of elements of several classes and a query class X to recognize from it, let us define $Positives$ as the number of elements belonging to the class X and $Negatives$ as the number of elements that does not belong to X. The precision, recognition rate (recall), and fall-out (false positive rate) measures are computed using the following equations:

$$\text{Precision} = \frac{|\text{True Positives}|}{(|\text{True Positives}| + |\text{False Positives}|)} \tag{14}$$

$$\text{Recognition Rate} = \text{Recall} = \frac{|\text{True Positives}|}{|\text{Positives}|} \tag{15}$$

$$\text{Fall} - \text{out} = \text{false posit. rate} = \frac{|\text{False Positives}|}{|\text{Negatives}|} \tag{16}$$

In Table 2, the recognition rates of the classification of this data set are shown, where the DTW approach is compared to the Zernike moments, GFD, ART, and DTW-cyclic, using the parameters defined above. One can see that with the proposed method, we reach a recognition rate of 96.9%, significantly improving the Zernike Moments (75.7%), ART (82.9%), GFD (78.8%), and DTW-cyclic (65.5%).

In Table 3, we show the experimental results with some different features that can be used for describing the symbols. In this experiment, our DTW approach is always used, but making use of different features described in the literature, specifically those proposed by Rath and Marti. In Table 3, we also show the recognition rates obtained using different numbers of regions (3, 4, and 5) in the feature extraction step of our approach. We can observe that Marti's features perform very well for the trebble and bass clefs (over 97% of recognition rate), but very poor with alto clefs (90%). Contrary, Rath's features achieve a good performance in alto clefs but have some problems with trebble clefs. Concerning our features, we can see that the division of the image

Table 3 Classification of clefs (%): recognition rates (RR.) of these three music classes using four models

Method	Rath	Marti	DTW 3z	DTW 4z	DTW 5z
nf	4	8	5	6	7
RR. trebble clef	95.8	97.3	96.7	96.3	96.2
RR. bass clef	96.1	97.6	96.5	96.3	96.5
RR. alto clef	96.5	90.1	94.3	96.1	97.1
Overall RR.	**96.1**	**95.0**	**95.8**	**96.2**	**96.6**
Overall precission	**96.5**	**94.6**	**96.2**	**96.6**	**96.9**
Overall fall-out	**2.0**	**2.6**	**2.2**	**2.0**	**1.8**

Overall recognition rate (RR.), precision, and fall-out of Rath's features, Marti's features and our DTW features, using 3, 4, and 5 regions (zones). nf = Number of features per column

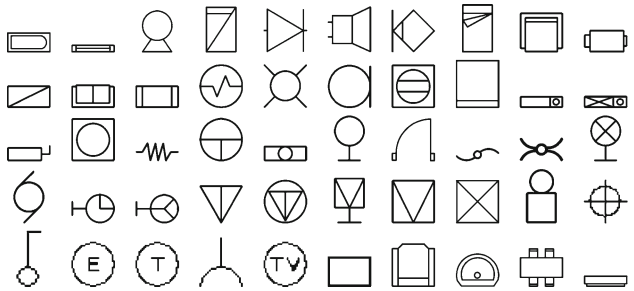

Fig. 11 The fifty selected representatives for the architectural database

in 3 regions does not provide enough discriminatory power for the high variability in alto clefs (we reach a recognition rate of 94.3%), while the recognition rate increases when the number of regions is increased, reaching a 97.1% with 5 regions. In addition, it is shown that the features we have used achieve a better overall recognition rate and precision (96.6 and 96.9%, respectively) in comparison with both of Marti's (95 and 94.6%) and Rath's ones (96.1 and 96.5%), with a lower fall-out (1.8% in comparison with 2.6% of Marti's and 2% of Rath's ones).

4.5 Architectural symbols data set

The architectural symbol data set is a benchmark database [35] comprising online and off-line instances from a set of 50 symbols drawn by a total of 21 users. Each user has drawn a total of 25 symbols and over 11 instances per symbol. Thus, the database (see examples in Fig. 2) consists on 7,465 individual instances, consisting of 50 symbols, each class with an average of 150 samples. In this database, the representative selected for each class (Fig. 11) corresponds to the printed symbol of the class, because both the printed and the hand-drawn symbols are quite similar.

The architectural symbol data set has been used to test the scalability of our method. In this experiment, we test the performance under an increasing number of classes.

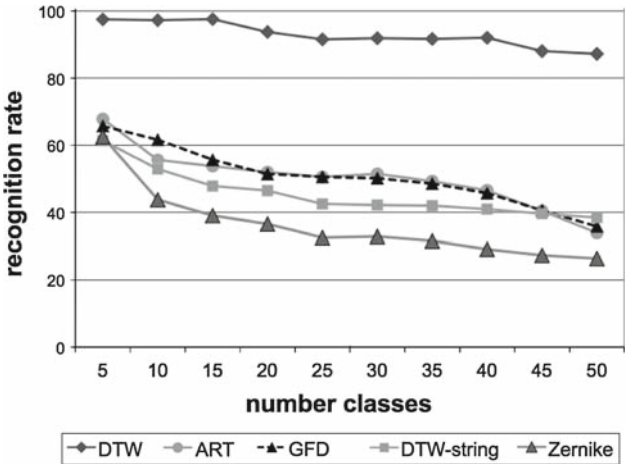

Fig. 12 Classification of architectural hand-drawn symbols: recognition rates using different number of classes

We have started the classification using the first five classes. Iteratively, five classes have been added at each step, and the classification has been repeated. The higher number of classes we introduce, the higher the confusion degree becomes among them. It is because of the elastic deformations inherent to hand-drawn strokes and the higher number of objects to distinguish. In Fig. 12, the recognition rates are presented, showing that our approach reaches significantly higher results than Zernike moments and the DTW-cyclic approach (87% in comparison with 26 and 38%, respectively). The performance of the Zernike moments, ART, GFD, and the DTW-cyclic decreases dramatically when increasing the confusion in terms of the number of classes (Zernike moments decreases from 62.5 to 26.4%, ART decreases from 67.8 to 33.9%, GFD decreases from 65.7 to 35.8%, and DTW-cyclic decreases from 61.3 to 38.4% with 50 classes), whereas our method is quite robust to the increasing of the number of different classes participating (from 97.5% with five classes decreases to 87.2% with 50 classes).

4.6 Discussions

Our DTW-based method has shown to be suitable for dealing with hand-drawn symbol recognition problems, being tolerant to elastic deformations, scale, and rotation. It has shown good performance with symbols with high variability (such as the music clefs data set), and also, shows a good scalability degree (see the results on the architectural symbols data set), outperforming the Zernike moments, ART, and GFD descriptors. The features proposed in our method also outperform the Rath's and Marti's ones.

An important point of our approach consists in the selection of the number of zones and the step orientations. Concerning the step orientation, a low value could help in decreasing the final matching cost, because more features (for each orientation) are computed, and the matching is more precise. However, one must take into account that if the number of angles W increases, the computational cost is also increased ($O(W^2N^2)$). Concerning the number of zones, they are used for defining the blurring degree allowed, in other words, a low number of zones will decrease the intra-class variability (but also the inter-class variability) and vice-versa. For this reason, the optimum number of zones will depend on each data set and will be a trade-off between inter-class and intra-class variability. A common way to look for the optimum number of zones is to use different values on a subset of the database and selecting the value that maximizes the recognition rate.

5 Conclusions

In this paper, we have presented a Dynamic Time Warping-based method for the description and classification of hand-drawn symbols. This approach is rotation and scale invariant, and robust to the deformations typical in hand-drawn symbols. The method proposed computes a column sequence of feature vectors for each orientation of the two symbols and computes the DTW distance, taking also into account their perpendicular alignment. Our method has been tested with two hand-drawn symbol databases (music and architectural) achieving high recognition rates. Comparison against some state-of-the-art descriptors shows the robustness and better performance of the proposed approach when classifying symbols with high variability in appearance, such as irregular deformations induced by hand-drawn strokes, low inter-class and high intra-class variabilities.

The main drawback is the high computational cost: even though the method proposed is $O(w^2N^2)$, which is remarkably lower than other existing 2D-DTW approaches (such as $O(N^{4N})$ and $O(N^3 9^N)$), it is still not fast enough for performing symbol recognition in large databases or even real-time symbol recognition systems. In this sense, further work can be focused on developing DTW-variations for decreasing the time complexity of the algorithm.

Acknowledgments We would like to thank Prof. J.M.Gregori for his help in accessing the historical archives, and V.Kilchherr and Dr. A.Schlapbach for providing support in the experiments. This work has been partially supported by the Spanish projects TIN2008-04998, TIN2009-14633-C03-03 and CONSOLIDER -INGENIO 2010 (CSD2007-00018).

References

1. Bartolini, I., Ciaccia, P., Patella, M.: Warp: accurate retrieval of shapes using phase of fourier descriptors and time warping distance. IEEE Trans. Pattern Anal. Mach. Intell. **27**(1), 142–147 (2005)

2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)
3. Bunke, H.: Attributed programmed graph grammars and their application to schematic diagram interpretation. IEEE Trans. Pattern Anal. Mach. Intell. **4**(6), 574–582 (1982)
4. Chan, K.F., Yeung, D.Y.: Mathematical expression recognition: a survey. Int. J. Document Anal. Recognit. **3**(1), 3–15 (2000)
5. Chou, K.S., Fan, K.C., Fan, T.I.: Peripheral and global features for use in coarse classification of Chinese characters. Pattern Recognit. **30**(3), 483–489 (1997)
6. Fonseca M.J., Pimentel C., Jorge J.A.: CALI: an online scribble recognizer for calligraphic interfaces. In: AAAI Spring Symposium on Sketch Understanding, pp. 51–58. (2002)
7. Fornés, A., Lladós, J., Sánchez, G.: Primitive segmentation in old handwritten music scores. In: Liu, W., Lladós, J. (eds.) Graphics Recognition: Ten Years Review and Future Perspectives, vol. 3926 of Lecture Notes in Computer Science, pp. 279–290. Springer (2006)
8. Garain, U., Chaudhuri, B.B.: Recognition of online handwritten mathematical expressions. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **34**(6), 2366–2376 (2004)
9. Golubitsky, O., Watt, S.M.: Online recognition of multi-stroke symbols with orthogonal series. International Conference on Document Analysis and Recognition, vol. 2, pp. 1265–1269 (2009)
10. Hse, H., Newton, A.R.: Sketched symbol recognition using Zernike moments. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, pp. 367–370 (2004)
11. Hu, N., Dannenberg, R.B., Tzanetakis, G.: Polyphonic audio matching and alignment for music retrieval. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 185–188. New Paltz, New York, October (2003)
12. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
13. Khotanzad, A., Hong, Y.H.: Invariant image recognition by Zernike moments. IEEE Trans. Pattern Anal. Mach. Intell. **12**(5), 489–497 (1990)
14. Kim, W.Y., Kim, Y.S.: A new region-based shape descriptor. Technical report, Hanyang University and Konan Technology (1999)
15. Kornfield, E.M., Manmatha, R., Allan, J.: Text alignment with handwritten documents. In Document Image Analysis for Libraries, pp. 195–209. IEEE Computer Society, Washington, DC, USA, (2004)
16. Kruskal, J.B., Liberman, M. : The symmetric time-warping problem: from continuous to discrete. In: Sankoff, D., Kruskal, J.B. (eds.) Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison., pp. 125–161. Addison-Wesley Publishing Co, Reading, MA (1983)
17. Khurshid, K., Faure, C., Vincent, N.: Fusion of word spotting and spatial information for figure caption retrieval in historical document images. 10th International Conference on Document Analysis and Recognition, vol. 1, pp. 266–270 (2009)
18. Levin, E., Pieraccini, R.: Dynamic planar warping for optical character recognition. In IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 149–152 (1992)
19. Liang, S., Sun, Z., Li, B.: Sketch retrieval based on spatial relations. Proceedings of International Conference on Computer Graphics, Imaging and Visualization, Beijing, China, pp. 24–29 (2005)
20. Liu, C.L.: Normalization-cooperated gradient feature extraction for handwritten character recognition. IEEE Trans. Pattern Anal. Mach. Intell., pp. 1465–1469 (2007)
21. Lladós, J., Valveny, E., Sánchez, G., Martí, E.: Symbol recognition: current advances and perspectives. In Lecture Notes in Computer Science, vol. 2390, pp. 104–128. Springer (2002)
22. Lladós, J., Martí, E., Villanueva, J.J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23(10), pp. 1137–1143, October (2001)
23. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. Int. J. Pattern Recogni. Artif. Intell. **15**, 65–90 (2001)
24. Mas, J., Jorge, J.A., Sánchez, G., Lladós, J.: Representing and parsing sketched symbols using adjacency grammars and a grid-directed parser. In: Ogier eds, J.M., Liu, W., Lladós, J. (eds.), Graphics Recognition: Recent Advances and New Opportunities, Lecture Notes in Computer Science, vol. 5046, pp. 176–187. Springer-Verlag (2008)
25. Miyao, H., Maruyama, M.: An online handwritten music symbol recognition system. Int. J. Document Anal. Recogni. **9**(1), 49–58 (2007)
26. Mokhtarian, F., Mackworth, A.K.: Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE Trans. Pattern Anal. Mach. Intell. **8**(1), 34–43 (1986)
27. Müller, S., Rigoll, G.: Engineering Drawing Database Retrieval Using Statistical Pattern Spotting Techniques. Graphics Recogni. Recent Adv. Lecture Notes Comput. Sci. **1941**, 246–255 (2000)
28. Niels, R., Vuurpijl, L.: Using dynamic time warping for intuitive handwriting recognition. Advances in Graphonomics, Proceedings of the 12th Conference of the International Graphonomics Society, pp. 217–221 (2005)
29. Orio, N., Schwarz, D.: Alignment of monophonic and polyphonic music to a score. In San Francisco International Computer Music Association, editor, Proceedings of the International Computer Music Conference, pp. 155–158, Havana, Cuba, September (2001)
30. Parker, J., Pivovarov, J., Royko, D.: Vector templates for symbol recognition. In: International Conference on Pattern Recognition, vol. 15, pp. 602–605 (2000)
31. Peng, B., Liu, Y., Wenyin, L., Huang, G.: Sketch recognition based on topological spatial relationship. In: Structural, Syntactic, and Statistical Pattern Recognition: Lecture Notes in Computer Science, vol. 3138, pp. 434–443. Springer-Verlag (2004)
32. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications inspeech recognition. Proc. IEEE **77**(2), 257–286 (1989)
33. Rath, T.M. Manmatha, R.: Word image matching using dynamic time warping. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 521–527 Madison, WI, June 18–20 (2003)
34. Rusiñol, M., Borràs, A., Lladós, J.: Relational indexing of vectorial primitives for symbol spotting in line-drawing images. Pattern Recogni. Lett. **31**(3), 188–201 (2010)
35. Sanchez, G., Valveny, E., Llados, J., Romeu, J. Mas, Lozano. N.: A platform to extract knowledge from graphic documents. Application to an architectural sketch understanding scenario. In: Dengel, A. Marinai, S., (eds.), Document Analysis Systems VI, Lecture Notes in Computer Science, vol. 3163, pp. 389–400, Florence—Italy, 2004, Springer-Verlag
36. Shi, Y., Li, H.Y., Soong, F.K.: A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. Ninth International Conference on Document Analysis and Recognition, vol. 2, 854–858, Sept. (2007)
37. Tabbone, S., Wendling, L.: Technical symbols recognition using the two-dimensional radon transform. In Proceedings of the 16th International Conference on Pattern Recognition, vol. 3, pp. 200–203 (2002)
38. Uchida, S., Sakoe, H.: A monotonic and continuous two-dimensional warping based on dynamic programming. In Proceedings of 14th International Conference on Pattern Recognition, vol. 1, pp. 521–524, (1998)

39. Uchida, S., Sakoe, H.: A survey of elastic matching techniques for handwritten character recognition. IEICE Trans. Inf. Syst. **E88-D**, 1781–1790 (2005)
40. Valveny, E., Marti, E.: Hand-drawn symbol recognition in graphic documents using deformable template matching and a bayesian framework. Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, pp. 239–242 (2000)
41. Vinciarelli, A., Bengio, S., Bunke, H.: Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Trans. Pattern Anal. Mach. Intell., pp. 709–720 (2004)
42. Vuori, V., Laaksonen, J., Oja, E., Kangas, J.: Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. Int. J. Document Anal. Recogni. **3**(3), 150–159 (2001)
43. Wilfong, G., Sinden, F., Ruedisueli, L.: On-line recognition of handwritten symbols. IEEE Trans. Pattern Anal. Mach. Intell. **18**(9), 935–940 (1996)
44. Xin, G., Cuiyun, L., Jihong, P., Weixin, X.: HMM based online hand-drawn graphic symbol recognition. In Proceedings of the 6th International Conference on Signal Processing, vol. 2, pp. 1067–1070 (2002)
45. Yang, S.: Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. IEEE Trans. Pattern Anal. Mach. Intell. **27**(2), 278–281 (2005)
46. Zhang, D.S., Lu, G.: Generic Fourier descriptor for shape-based image retrieval. In Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 1, pp. 425–428 (2002)
47. Zhang, W., Liu, W.: A new syntactic approach to graphic symbol recognition. International Conference on Document Analysis and Recognition vol. 1, pp. 516–520 (2007)
48. Zhang, W., Liu, W., Zhang, K.: Symbol recognition with kernel density matching. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2020–2024 (2006)