Recognition of Pen-Based Music Notation: the HOMUS dataset

Jorge Calvo-Zaragoza and Jose Oncina
Department of Software and Computing Systems
University of Alicante
Spain
Email: {jcalvo,oncina}@dlsi.ua.es

***Abstract*—A profitable way of digitizing a new musical composition is by using a pen-based (online) system, in which the score is created with the sole effort of the composition itself. However, the development of such systems is still largely unexplored. Some studies have been carried out but the use of particular little datasets has led to avoid objective comparisons between different approaches. To solve this situation, this work presents the Handwritten Online Musical Symbols (HOMUS) dataset, which consists of 15200 samples of 32 types of musical symbols from 100 different musicians. Several alternatives of recognition for the two modalities –online, using the strokes drawn by the pen, and offline, using the image generated after drawing the symbol– are also presented. Some experiments are included aimed to draw main conclusions about the recognition of these data. It is expected that this work can establish a binding point in the field of recognition of online handwritten music notation and serve as a baseline for future developments.**

I. Introduction

Composing music with pen and paper is still a common procedure. However, there may be several reasons for exporting a music score to a digital format: storage, distribution and reproduction; using its information in the search of musical pieces; grouping of styles and detection of plagiarism; or for building digital libraries. Conventional digital score editors put musical symbols on a score by using *point and click* actions with the mouse. These tools represent a tedious effort for the user, leading to consume a lot of time. The use of digital instruments seems a more comfortable alternative. Digital instruments (such as a MIDI piano) can be connected directly to the computer and transfer the information while playing the musical piece. However, this type of transcription is not error-free and rarely catch all the nuances that may contain a score. Moreover, the music sheet can be scanned in order to use an automatic score transcription tool –commonly referred as Optical Music Recognition (OMR) systems [1]–. This option represent an effortless alternative for the user. Unfortunately, OMR systems are far from achieving accurate transcriptions, especially for handwritten scores [2]. Thus, the transcription has to be corrected afterwards.

Although one of the above methods can be used, it is more profitable digitizing the score at the same time the composer writes. In this way, the score is digitized with the sole effort of the composition itself. With an online transcription system, many of the problems above discussed can be avoided, plus additional advantages (e.g., the ability to quickly reproduce the current composition). Furthermore, recognition of this kind of musical symbols could have use in other contexts. For instance, it is feasible to think of a scenario in which an OMR system allows corrections using a digital pen, rather than having to use the conventional mechanism of a score editor. This approach has been already applied to Handwritten Text Recognition [3].

Some previous studies have been carried out but this field still remains largely unexplored. One of the major absences is a dataset that serve as reference for research. All the previous works have worked with its own dataset and its own set of musical symbols. Therefore, comparative studies to know which approaches perform better than others have not been conducted so it is still unclear what is the current status of the research. The present work aims to set a reference point for research on recognition of online handwritten musical symbols. To this end, a large dataset is provided for free access [1], covering the most used symbols in the composition of musical scores. To establish the first baseline, experimentation with well-known pattern recognition algorithms is presented so that more information about the dataset can be known such as the difficulty of the recognition task or which techniques seem more promising. It is also expected that the results can serve as baseline for future comparisons and developments.

The rest of the paper is structured as follows: Section II describes the nature of the recognition of online handwritten music notation. The description of the dataset is shown in Section III. Section IV presents some baseline techniques for this dataset. Experiments are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Recognition of Pen-Based Handwritten Music Notation

Over decades, much research has been devoted to the development of friendly music score editors. Despite all these efforts, there is still no satisfactory solution. The emergence of tablet computer devices has open new avenues to approach this problem. With these devices, a musician can compose its music on a digital score using an electronic pen and have it effortlessly digitized.

The recognition of online (or pen-based) handwritten music notation task is defined as recognition of musical symbols at the time they are being written. The great variability in the manner of writing the musical symbols is the main difficulty to overcome. Figure 1 shows some examples of handwritten musical symbols from different musicians.

[1]The dataset is available at `http://grfia.dlsi.ua.es/homus/`

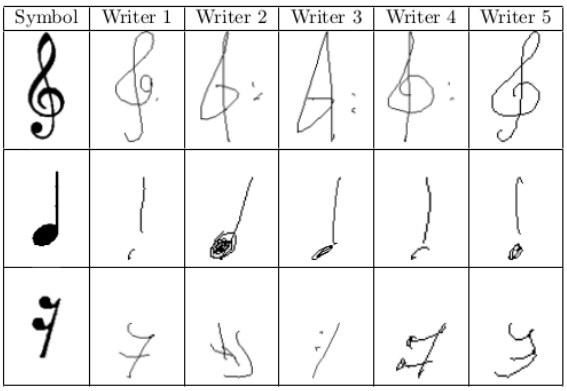

Fig. 1. Some examples of variability in handwritten musical symbols.

This variability is also a problem in OMR systems, but this scenario offers important advantages with respect to them: the staff lines (one of the main issues in offline OMR systems) do not interfere in the recognition since they are handle by the underlying system, the symbol detection could be intrinsically performed somehow, and the information about how the strokes are drawn is available.

These strokes –considered as the shape between pen-down and pen-up actions– produce an ordered set of points, which indicate the path followed by the pen. Similarly, each symbol can be drawn by one or more strokes. But not only this information can be extracted. An image of the shape itself can also be used for the classification (as it would be done in offline recognition). This modality gives another perspective of the symbol and it is more robust against the speed of the user, the order followed to draw a symbol and the number of strokes used.

A. Background

The first systems for pen-based recognition of musical scores were based on the use of simple gestures. This is the case of *Presto* system [4], which received as input short gestures that were generally mnemonic of the music symbols. These gestures were processed and translated to the actual musical symbols. With the same idea, Polácek et al. [5] created a new gesture alphabet especially designed for its use in low-resolution devices. The main drawback of these approaches is that they require an adaptation of the user to the gesture alphabet recognized by the system. Subsequently, there were other works that allowed writing symbols in its conventional manner. Miyao and Maruyama [6] based its system on the recognition of primitives (lines, circles, arcs, etc.), using information both the stroke path and the shape drawn. After the recognition, these primitives are combined to reconstruct the musical symbols. A similar approach was used in [7], in which document spatial structures were defined and combined with context-free grammars. However, depending on the musician writing, a musical symbol may consist of different primitives, so that the rules to rebuild the symbols lack the robustness needed to handle the different writing styles. Moreover, systems that have as their objective the recognition of complete musical symbol can also be found. George [8] used the images generated by the digital pen to learn an Artificial Neural Network (ANN) to recognize the symbols. Lee et al. [9] proposed the use of Hidden Markov Models (HMM) for recognition of some of the most common musical symbols using different features of the shape drawn by the pen. These studies have shown that the complete recognition of symbols written in the natural form of music is feasible.

TABLE I. FEATURES OF DATASETS USED IN PREVIOUS WORKS.

Work	Classes	Users	Data
George	20	25	4188 images
Miyao and Maruyama	12	11	13801 strokes
Lee et al.	8	1	400 symbols
Our dataset	32	100	15200 symbols

The recognition of online handwritten music notation is still a novel field so it is not yet established guidelines about which types of algorithms perform better. Aforementioned works have performed experiments that were only focused on finding the optimal parameters of the specific algorithm used. Each of them used its own dataset, its own set of musical symbols and its own nature of the input (see Table I), so it is unclear what dataset must be used to evaluate the performance of new approaches. This has hitherto led to a lack of comparative experiments to assess which of the proposed algorithms perform better in this context. To provide a solution to this problem, this work presents the HOMUS dataset, described in the next section.

III. THE HANDWRITTEN ONLINE MUSICAL SYMBOLS DATASET

This section presents the Handwritten Online Musical Symbols (HOMUS) dataset. The objective is to provide a reference corpus for research on the recognition of online handwritten music notation. The dataset is available at `http://grfia.dlsi.ua.es/homus/`.

Analyzing previous works, it was observed that most of them only took into account a small set of the possible musical symbols. In addition, it is important to stress that each musician has its own writing style, as it occurs in handwritten text. Increasing both the set of musical symbols and the number of different writing styles is advisable if reliable results about the recognition of online handwritten music notation are pursued.

Following this way, the HOMUS was built by 100 musicians from the *Escuela de Educandos Asociación Musical l'Avanç* (El Campello, Spain) and *Conservatorio Superior de Música de Murcia "Manuel Massotti Littell"* (Murcia, Spain) music schools, among whom were both music teachers and advanced students. In order to cover more scenarios, some of them were experienced in handwritten music composition while other have few composition experience. Musicians were encouragingly asked to draw the symbols trying not to do it in a perfect manner, but in its own, particular style (which is reflected in the variability shown in Fig. 1). Each of them were asked to draw four times the 32 classes listed in Table II, which has resulted in 15200 samples spread over 38 templates [2]. Each sample of the dataset contains the label and the strokes composing the symbol. These strokes consists of a set of points relative to a coordinate center. Storing the data in this way allows covering all the possibilities considered: the image can be generated from the strokes, every single stroke can be

[2]The eighth, sixteenth, thirty-second, and sixty-fourth note symbols are written twice: right and inverted.

TABLE II. TYPES OF MUSICAL SYMBOLS IN THE HOMUS DATASET.

Note	whole, half, quarter, eighth, sixteenth, thirty-second, sixty-fourth
Rest	whole/half, quarter, eighth, sixteenth, thirty-second, sixty-fourth
Accidentals	flat, sharp, natural, double sharp
Time signatures	common time, cut time, 4-4, 2-2, 2-4, 3-4, 3-8, 6-8, 9-8, 12-8
Clef	G-clef, C-clef, F-clef
Others	dot, barline

extracted easily, and each individual symbol remains isolated. Since the pitch of the notes is based on its position over the staff, it is unnecessary to detect it in the classification, but it may be assigned in a post-processing stage.

It should be noted that not all musical symbols appear in the dataset. Less relevant symbols such as accidentals, ornaments or instrument-specific notation were left out although they could be added to the score with another mechanism (e.g., via a contextual menu). There are other symbols that can not be present because of their unfixed length (such as ties or slurs) for which an alternative mechanism of addition can also be found.

To create the dataset a *Samsung Galaxy Note 10.1* device was used and symbols were written using the stylus *S-Pen*. This device was chosen among the standalone friendly options because of its optimality to work with an e-pen. The device has a resolution of 1280×800 (149 ppi) and a sampling rate of 16 ms (60 fps). An application that request musical symbols to be drawn on an empty staff was developed. The staff was composed of five parallel lines with a line thickness of 3 and an equal staff line spacing of 14. These two values are provided as a reference for possible rescaling since they are the common features for this purpose in OMR systems [10].

In addition to the dataset, this paper is intended to provide a baseline of the classification rate that can be achieved. Some basic techniques to recognize HOMUS samples are described in the next section.

IV. BASELINE TECHNIQUES

In this section, some techniques for the recognition of the samples contained in the HOMUS dataset are presented. The goal is not to achieve high success rates, but provide some notions about the classification of the symbols. It is also expected that experiments identify the most promising techniques to recognize this kind of data and the results can be used as baseline to compare future developments.

The dual nature of the data –using the strokes and using the image– leads us to explore both ways in the classification of the symbols. Classification techniques for each of these modalities are presented in the following subsections.

A. Online Techniques

The online recognition modality uses the strokes made by the pen. These strokes provide information about how the shape has been generated segment by segment. This modality takes advantage of the local information, expecting that a particular musical symbol follows similar paths. Depending on the type of musical symbol and the pace of the user, a greater or lower number of points will be generated. Therefore, each sample has a different dimension. Due to this, most of the conventional techniques based on equal-sized feature vectors can not be applied. For this reason, we will restrict ourselves to the use of the Nearest Neighbor (NN) technique and Hidden Markov Models (HMM).

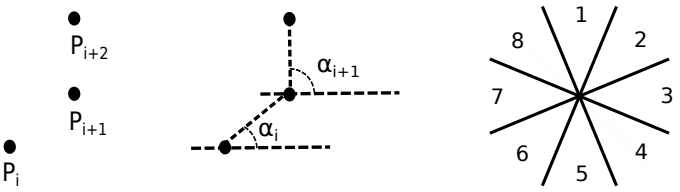

Fig. 2. FCC based on the angle between consecutive points.

1) Nearest Neighbor: Let $X = (x_1, \ldots, x_n)$ be a set of labeled samples and let $x' \in X$ be the sample that minimizes a dissimilarity measure $d(x, x')$ to a test point x. The NN rule [11] assigns to x the label associated with x'. The natural extension of this rule is to use the k-nearest samples (k-NN) and assign the most frequent label. The performance of this rule is strongly related to the dissimilarity measure $d(x, x')$ utilized. Two alternatives are presented in the following lines: Edit Distance with Freeman Chain Codes (FCC) and Dynamic Time Warping (DTW).

Given two strings, the edit distance (or Levenshtein distance) [12] is the minimum number of edit operations –usually insertion, deletion and substitution– to convert one string into another. To use this distance over the samples of the HOMUS, the set of points that represents a musical symbol has to be converted into a string. Codification based on Freeman Chain Code (FCC) [13] is applied. FCC is a typical method to build strings from image contours. It converts each pair of pixels into one code in function of the neighboring direction. In this case, instead of a contour we have a set of points that are not continuous (because of the device sampling rate). This situation can be approached in many ways. Between each pair of points a line that connects them can be interpolated. Thus we can establish a continuous path and applying the conventional FCC afterwards. Moreover, each pair of points can be replaced by a code based on the angle they form (see Fig. 2). These two approaches (FCC and FCC based on angle) will be evaluated experimentally. For symbols with multiple strokes, a specific code is concatenated at the end of each stroke.

On the other hand, DTW is a technique for measuring the dissimilarity between two time signals which may be of different durations. It was firstly used in speech recognition [14] although its use has widely extended to other fields [15], [16]. Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_m)$ be two time series, of length n and m respectively. DTW(i,j) is defined recursively as

$$\begin{cases} 0, & j = 0 \wedge i = 0 \\ \infty, & j = 0 \wedge i > 0 \\ \infty, & i = 0 \wedge j > 0 \\ d(x_i, y_i) + \min \begin{cases} DTW(i-1, j) \\ DTW(i, j-1) \\ DTW(i-1, j-1) \end{cases}, & otherwise \end{cases} \quad (1)$$

and therefore, DTW(x,y) = DTW(n,m). In our case, x_i and y_i are points in a 2-dimensional space. Hence, the distance $d(x_i, y_j)$ is the Euclidean distance between two points. The algorithm is implemented using a dynamic programming scheme, reducing the complexity to $O(nm)$. Details about the intrinsic operation of the algorithm can be found in [14].

2) Hidden Markov Models: Hidden Markov Models (HMM) [17] are statistical models that define an unobservable Markov process generating an observable output sequence. They have been successfully used in online handwritten recognition during the last years [18], [19]. In our work, a *continuous left-to-right* topology is used and the models are trained with the Baum-Welch algorithm [20]. Both the number of states and the number of Gaussian densities in the mixtures are adjusted in preliminary experiments.

Feature extraction is performed as described in the work of Kian et al. [9], which obtained good results for online music symbol recognition.

B. Offline Techniques

After drawing the symbol, an image can be obtained by creating lines between pair of consecutive points. After this, the lines are dilated to simulate a thickness of 3 as used to collect the samples. The information contained in these images provide a new perspective on the recognition and it does not overlap with the nature of the online recognition. The advantage of this representation is that it is robust against different speeds or different orders when writing the symbol.

The baseline showed here is inspired by the work of Rebelo et al. [21] on offline musical symbol recognition. The algorithms considered are k-Nearest Neighbor, Artificial Neural Network, Support Vector Machines and Hidden Markov Models. The images are resized to 20×20 and no feature extraction is performed (except for Hidden Markov Models).

1) k-Nearest Neighbor: The k-Nearest Neighbor (k-NN) rule, explained in the previous subsection, can also be used for recognition from images. In this case, a 400-dimensional vector with real values is received as input. To measure the dissimilarity between two samples, the Euclidean distance is used. Some different values for the parameter k will be evaluated experimentally (1, 3 and 5).

2) Artificial Neural Networks: Artificial Neural Networks (ANN) emerged as an attempt to mimic the operation of the nervous system to solve machine learning problems. An ANN comprises a set of interconnected neurons following a certain topology. Further details about ANN can be found in [22].

The topology of a neural network can be quite varied. For this work, the common neural network called Multi-Layer Perceptron (MLP) is used. This topology was also used for the same purpose in the work of George [8]. This kind of networks can be trained with the backpropagation algorithm [23]. The number of hidden states was fixed to 200.

3) Support Vector Machines: Support Vector Machines (SVM) is a supervised learning algorithm developed by Vapnik [24]. It seeks for a hyperplane

$$h(x) = w^T x + b = 0 \tag{2}$$

which maximizes the separation (margin) between the hyperplane and the nearest samples of each class (support vectors). Among the alternatives to extend the algorithm for multi-class problems, the *one-vs-one* scheme is used here.

SVM relies on the use of a Kernel function to deal with non-linearly separable problems. In this work, two kernel functions will be considered: radial basis function (RBF) kernel (Eq. 3) and polynomial (Poly) kernel (Eq. 4).

$$K(x, y) = e^{-\gamma \cdot \|x-y\|^2} \tag{3}$$

$$K(x, y) = \langle x, y \rangle^n \tag{4}$$

The training of the SVM is conducted by the Sequential Minimal Optimization (SMO) algorithm [25].

4) Hidden Markov Models: HMM are used here as explained for the online data. In this case, resizing and feature extraction are performed like in the work of Pugin [26].

V. Experimentation

The experimental part of this work focuses on providing the first classification results for the HOMUS dataset. In this way, we try to show what aspects of these data seem more appropriate or what are the main challenges to recognize the different musical symbols. To this end, two experiments are presented in this section. The first experiment is carried out to assess if the algorithms can detect the symbols regardless the particular style of each musician. The second experiment is aimed to analyze the accuracy of the algorithms when samples of the same user have been presented during the training stage. Next subsections describe these experiments.

A. User-independent experiment

In this experiment we aim to assess the difficulty of recognizing symbols from an unknown user. The samples of each musician are isolated from the whole dataset and used as test set (100 sets). Then, a 100-fold cross validation is conducted using a common $0-1$ loss function. The error rates obtained after applying the algorithms described in Section IV are shown in Fig. 3 (user-independent columns).

As seen in the results, algorithms are not very reliable in this scenario since all of them obtain error rates higher than 15 %. DTW obtains the lowest error rate (15.2 %). Among the offline techniques, SVM with RBF kernel provides the best error rate (26 %). To measure the significance of the results, a *Wilcoxon* statistical test was performed using KEEL software [27] (see Table III). It can be seen that DTW achieves significantly better results than other techniques.

B. User-dependent experiment

The latter experiment is focused on assessing how the classification results are affected when samples of the same musician are found in the training set. Each musician is divided into four sets and each one is used as a fold for a cross-validation experiment. To build the training set, two alternatives can be used: (1) using only the rest of the samples of the same musician (user set), and (2) using the rest of the dataset, including the remaining samples of the same musician

TABLE III. SUMMARY OF THE WILCOXON TEST FOR USER-INDEPENDENT EXPERIMENT. ●= THE METHOD IN THE ROW IMPROVES THE METHOD OF THE COLUMN. ○= THE METHOD IN THE COLUMN IMPROVES THE METHOD OF THE ROW. UPPER DIAGONAL OF LEVEL SIGNIFICANCE $\alpha = 0.9$, LOWER DIAGONAL LEVEL OF SIGNIFICANCE $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-	●	●	●	●	●	●	●	●	●	●
String (2)	○	-	●	●	●	●	●	●	●	●	●
Angle (3)	○	○	-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	○	●	●	●	●	●
MLP (5)	○	○	○	○	-	○	○	○	○	○	○
RBF (6)	○	○	○	●	●	-	●	●	●	●	●
Poly (7)	○	○	○	○	●	○	-	●	●	●	●
1NN (8)	○	○	○	○	●	○	○	-	○	●	●
3NN (9)	○	○	○	○	●	○	○	●	-	●	
5NN (10)	○	○	○	○	●	○	○	○	○	-	
HMM_{off} (11)	○	○	○	○	●	○	○	○			-

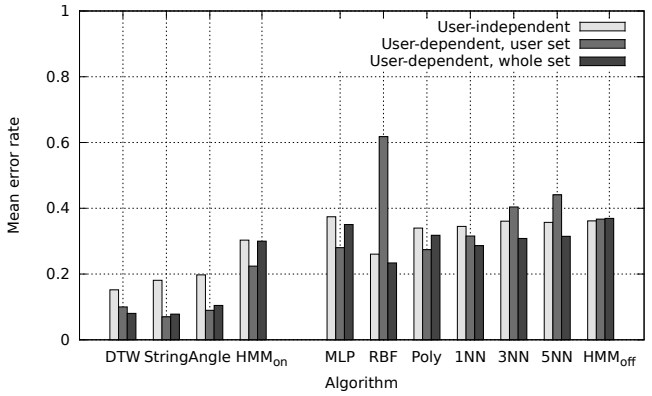

Fig. 3. Mean error rate of classification experiments. **String**: Freeman Chain Code, **Angle**: Freeman Chain Code based on angles, **DTW**: Dynamic Time Warping, $\mathbf{HMM_{on}}$: Hidden Markov Models with online features, **MLP**: Multi-Layer Perceptron, **Poly**: Support Vector Machine with Polynomial kernel, **RBF**: Support Vector Machine with Radial Basis Function kernel, **k-NN**: k-Nearest Neighbor using images, $\mathbf{HMM_{off}}$: Hidden Markov Models with offline features.

(whole set). This two options are confronted experimentally in a 400-fold (four per musician) cross validation. Figure 3 (user-dependent columns) show the results of this experiment, which is measure using the $0-1$ loss function as well.

Algorithms using the online nature of the data have the best performance while those exploiting offline modality still have higher error rates. Conventional FCC has reported the best error rate, on average (7 %). Regarding the two ways of building the training set, there are no clear trend in the results. Some algorithms have improved when using the whole dataset such as NN family and, especially, SVM with a RBF kernel (from 61 % to 23 %) because of its poor performance with few training data. However, in other algorithms, the error rate hardly varies or even rises, as in the case of SVM with a Polynomial kernel, the MLP or HMM with online features. The Wilcoxon statistical tests for these experiments are shown in Table IV and V. If each modality is seen as a whole, algorithms that work with the online data, except for HMM, achieve significantly better results than the others.

Comparing these results with those obtained in the previous experiment we can conclude that including samples of the same user during the training set can remarkably improve the performance of some algorithms. For instance, FCC has improved considerably its performance from 18 % to 7 % of error rate. Depending on the algorithm used, it is more

TABLE IV. SUMMARY OF THE WILCOXON TEST FOR USER-DEPENDENT (USER SET) EXPERIMENT. ●= THE METHOD IN THE ROW IMPROVES THE METHOD OF THE COLUMN. ○= THE METHOD IN THE COLUMN IMPROVES THE METHOD OF THE ROW. UPPER DIAGONAL OF LEVEL SIGNIFICANCE $\alpha = 0.9$, LOWER DIAGONAL LEVEL OF SIGNIFICANCE $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-			●	●	●	●	●	●	●	●
String (2)		-		●	●	●	●	●	●	●	●
Angle (3)			-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	●	●	●	●	●	●
MLP (5)	○	○	○	○	-	○			○	○	●
RBF (6)	○	○	○	○	●	-	●	●	●	●	○
Poly (7)	○	○	○	○		○	-	○	○	○	●
1NN (8)	○	○	○	○		○	●	-	○	○	●
3NN (9)	○	○	○	○	●	○	●	●	-		○
5NN (10)	○	○	○	○	●	○	●	●		-	○
HMM_{off} (11)	○	○	○	○	○	●	○	○	●	●	-

TABLE V. SUMMARY OF THE WILCOXON TEST FOR USER-DEPENDENT (WHOLE SET) EXPERIMENT. ●= THE METHOD IN THE ROW IMPROVES THE METHOD OF THE COLUMN. ○= THE METHOD IN THE COLUMN IMPROVES THE METHOD OF THE ROW. UPPER DIAGONAL OF LEVEL SIGNIFICANCE $\alpha = 0.9$, LOWER DIAGONAL LEVEL OF SIGNIFICANCE $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-			●	●	●	●	●	●	●	●
String (2)		-		●	●	●	●	●	●	●	●
Angle (3)			-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	○	●	○		●	●
MLP (5)	○	○	○	○	-	●	●	●	●	●	●
RBF (6)	○	○	○	●	○	-	○	○	○	○	●
Poly (7)	○	○	○	○	○	●	-				●
1NN (8)	○	○	○	●	○	●		-	○		●
3NN (9)	○	○	○		○	●		●	-		●
5NN (10)	○	○	○	○	○	●				-	●
HMM_{off} (11)	○	○	○	○	○	○	○	○	○	○	-

convenient to do it with the rest of the dataset or only with the remaining samples of the same user. Algorithms that exploit the online modality of the data, except for the HMM, have shown a significantly better performance in both experiments. Specifically, DTW has proven to be the best technique since it improves significantly the results of other algorithms in the user-independent experiment and no one is significantly better in the user-dependent experiments. HMM deserves further consideration because its performance is closely linked to feature extraction. In any case, in this work we focused on features used in previous studies for the same task.

VI. CONCLUSIONS

The work presented here aims to become a first point of reference for recognition of online handwritten music notation. This process is focused on recognizing musical symbols that are drawn on a digital score using a friendly tablet device and an electronic pen. In this way, musicians can digitize their compositions without resorting to conventional music score editors.

Some previous studies that have worked on this issue have been presented. However, all of them used their own corpus, so there is still a lack of comparative experiments that indicate which algorithms are better for this task. To solve this problem, this paper has presented the HOMUS (Handwritten Online Musical Symbols) dataset. This dataset contains 15200 samples of musical symbols from 100 expert musicians. Within this set, 32 different types of musical symbols can be found. It is expected that the dataset provides sufficient samples so that the results depend on the techniques used for classification.

To establish the first baseline, experiments with well-known pattern recognition algorithms have been carried out. FCC,

DTW and HMM have been used to take advantage of the online nature of these data while k-NN, SVM, ANN and HMM have been utilized to classify samples from the offline modality (image). Two experiments were conducted to better understand this dataset and draw the first conclusions on the classification of these symbols. The first experiment consists in measuring the difficulty of recognizing a symbol when it comes from an unknown musician (user-independent). In the second experiment, samples of the same musician are included in the training set (user-dependent). Results showed that recognizing symbols from unseen styles presents the main difficulty. Error rates of the user-independent experiment among 32 classes did not dropped below 15 % in any of the algorithms considered. Algorithms that exploit the online nature of the data has proven to be the most promising for the classification task, achieving results that improve the performance of those which use the offline modality. Considering all the experiments, DTW has shown the best performance. Nevertheless, results showed room for improvement.

These results has also led to the conclusion that a competitive system will need samples of the actual user. This scenario is feasible in real-world cases. The user can be asked to perform a training phase before using the system, in which he writes all the musical symbols with his own style. This extra effort can prevent a large number of classification errors that must be posteriorly corrected. Forcing the user to perform this phase can be actually seen as a way to minimize the human effort throughout the entire process. The user can also provide his writing style transparently by means of corrections where a misclassification is produced (user adaptation techniques).

As future work, the main challenge is to extend this work to recognize entire music scores.

Acknowledgment

This work was partially supported by a FPU fellowship (AP2012-0939) from the Spanish Ministerio de Educacin, Cultura y Deporte, the Spanish CICyT through the project TIN2009-14205-C04-C1 and Consejería de Educación de la Comunidad Valenciana through project PROMETEO/2012/017.

The authors are grateful to all the people who collaborate in the creation of the dataset.

References

[1] D. Bainbridge and T. Bell, "The Challenge of Optical Music Recognition," *Language Resources and Evaluation*, vol. 35, pp. 95–121, 2001.

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. Marcal, C. Guedes, and J. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, pp. 1–18, 2012.

[3] D. Martin-Albo, V. Romero, and E. Vidal, "Interactive off-line handwritten text transcription using on-line handwritten text as feedback," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 2013, pp. 1280–1284.

[4] J. Anstice, T. Bell, A. Cockburn, and M. Setchell, "The design of a pen-based musical input system," in *Sixth Australian Conference on Computer-Human Interaction, 1996. Proceedings.*, 1996, pp. 260–267.

[5] O. Poláček, A. J. Sporka, and P. Slavík, "Music alphabet for low-resolution touch displays," in *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, ser. ACE '09. New York, NY, USA: ACM, 2009, pp. 298–301.

[6] H. Miyao and M. Maruyama, "An online handwritten music symbol recognition system," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 1, pp. 49–58, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10032-006-0026-9

[7] S. Macé, Éric Anquetil, and B. Couasnon, "A generic method to design pen-based systems for structured document composition: Development of a musical score editor," in *Proceedings of the First Workshop on Improving and Assesing Pen-Based Input Techniques*, Edinghburg, 2005, pp. 15–22.

[8] S. E. George, "Online pen-based recognition of music notation with artificial neural networks," *Comput. Music J.*, vol. 27, no. 2, pp. 70–79, Jun. 2003.

[9] K. C. Lee, S. Phon-Amnuaisuk, and C.-Y. Ting, "Handwritten music notation recognition using hmm – a non-gestural approach," in *International Conference on Information Retrieval Knowledge Management, (CAMP), 2010*, 2010, pp. 255–259.

[10] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 753–766, 2008.

[11] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[12] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Tech. Rep. 8, 1966.

[13] H. Freeman, "On the encoding of arbitrary geometric configurations," *Electronic Computers, IRE Transactions on*, vol. EC-10, no. 2, pp. 260–268, 1961.

[14] H. Sakoe and S. Chiba, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Dynamic programming algorithm optimization for spoken word recognition, pp. 159–165.

[15] M. Faundez-Zanuy, "On-line signature recognition based on vq-dtw," *Pattern Recognition*, vol. 40, no. 3, pp. 981 – 992, 2007.

[16] B. Hartmann and N. Link, "Gesture recognition with inertial sensors and optimized dtw prototypes," in *IEEE International Conference on Systems Man and Cybernetics (SMC), 2010*, 2010, pp. 2102–2109.

[17] Z. Ghahramani, "Hidden markov models." River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002, ch. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42. [Online]. Available: http://dl.acm.org/citation.cfm?id=505741.505743

[18] L. Hu and R. Zanibbi, "Hmm-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011, pp. 457–462.

[19] S. Azeem and H. Ahmed, "Combining online and offline systems for arabic handwriting recognition," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 3725–3728.

[20] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA, USA: MIT Press, 1997.

[21] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols: A comparative study," *Int. J. Doc. Anal. Recognit.*, vol. 13, no. 1, pp. 19–31, Mar. 2010.

[22] D. Graupe, *Principles of Artificial Neural Networks*, 2nd ed. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: foundations of research," J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, ch. Learning representations by back-propagating errors, pp. 696–699.

[24] V. N. Vapnik, *Statistical learning theory*, 1st ed. Wiley, Sep. 1998.

[25] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast training of support vector machines using sequential minimal optimization, pp. 185–208.

[26] L. Pugin, "Optical music recognitoin of early typographic prints using hidden markov models," in *ISMIR*, 2006, pp. 53–56.

[27] J. Alcal-Fdez, L. Snchez, S. Garca, M. Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernndez, and F. Herrera, "Keel: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00500-008-0323-y