Pattern Recognition Approach for Music Style Identification Using Shallow Statistical Descriptors

Pedro J. Ponce de León and José M. Iñesta

***Abstract*—In the field of computer music, pattern recognition algorithms are very relevant for music information retrieval applications. One challenging task in this area is the automatic recognition of musical style, having a number of applications like indexing and selecting musical databases. From melodies symbolically represented as digital scores (standard musical instrument digital interface files), a number of melodic, harmonic, and rhythmic statistical descriptors are computed and their classification capability assessed in order to build effective description models. A framework for experimenting in this problem is presented, covering the feature extraction, feature selection, and classification stages, in such a way that new features and new musical styles can be easily incorporated and tested. Different classification methods, like Bayesian classifier, nearest neighbors, and self-organizing maps, are applied. The performance of such algorithms against different description models and parameters is analyzed for two particular musical styles, jazz and classical, used as an initial benchmark for our system.**

***Index Terms*—Bayesian classifier, music style classification, nearest neighbors, self-organizing maps (SOMs).**

I. Introduction

COMPUTER music research is an emerging area for pattern recognition and machine learning techniques to be applied. The content-based organization, indexing, and exploration of digital music databases (digital music libraries), where digitized (MP3), sequenced [musical instrument digital interface (MIDI)], or structurally represented (XML) music can be found, is known as music information retrieval (MIR). Efforts to standarize the descriptions for content-based search and retrieval of multimedia documents like MPEG-7 are already being developed.

One of the problems to solve in MIR is the modelization of music style. The computer could be trained to recognize the main features that characterize music genres so as to look for that kind of music over large musical databases. The same scheme is suitable to learn stylistic features of composers or even model a musical taste for users. Another application of such a system can be its use in cooperation with automatic composition algorithms to guide this process according to a given stylistic profile.

A number of recent papers explore the capabilities of machine learning methods to recognize music style. Pampalk *et al.* [1] use self-organizing maps (SOMs) to pose the problem of organizing music digital libraries according to sound features of musical themes, in such a way that similar themes are clustered, performing a content-based classification of the sounds. Whitman *et al.* [2] present a system based on neural networks and support vector machines able to classify an audio fragment into a given list of sources or artists. In [3], a neural system to recognize music types from sound inputs is described. An *emergent* approach to genre classification is used in [4], where a classification emerges from the data without any *a priori* given set of styles. The authors use co-ocurrence techniques to automatically extract musical similarity between titles or artists. The sources used for classification are radio programs and databases of compilation CDs.

Other works use music data in symbolic form (most MIDI data) to perform style recognition. Dannenberg *et al.* [5] use a naive Bayes classifier, a linear classifier, and neural networks to recognize up to eight moods (genres) of music, such as lyrical, frantic, etc. Thirteen statistical features derived from MIDI data are used for this genre discrimination. In [6], pitch features are extracted both from MIDI data and audio data and used separately to classify music within five genres. Pitch histograms regarding the tonal pitch are used in [7] to describe blues fragments of the saxophonist Charlie Parker. Also, pitch histograms and SOMs are used in [8] for musicological analysis of folk songs. Other researchers use sequence-processing techniques like hidden Markov models [9] and universal compression algorithms [10] to classify musical sequences.

Stamatatos and Widmer [11] use stylistic performance features and the discriminant analysis technique to obtain an ensemble of simple classifiers that work together to recognize the most likely music performer of a piece given a set of skilled candidate pianists. The input data are obtained from a computer-monitored piano, capable of measuring every key and pedal movement with high precision.

Compositions from five well-known Eighteenth Century composers are classified in [12] using 20 style features, most of them being counterpoint characteristics, and several supervised learning methods, such as k-means clustering, k-nearest-neighbor, and decision trees. This paper offers some conclusions about the differences between composers discovered by the different learning methods.

In [13], the ability of grammatical inference methods for modeling musical style is shown. A stochastic grammar for each musical style is inferred from examples, and those grammars are used to parse and classify new melodies. The authors also discuss about the encoding schemes that can be used to achieve the best recognition result. Other approaches like multilayer feedforward neural networks [14] have been used to classify musical style from symbolic sources.

Manuscript received March 8, 2004; revised July 23, 2004. This work was supported by the Spanish CICYT through project TIC2003-08496-C04, supported in part by EU ERDF, and Generalitat Valenciana GV043-541.

The authors are with the Department of Software and Computing Systems, University of Alicante, 03080 Alicante, Spain (e-mail: pierre@dlsi.ua.es; inesta@dlsi.ua.es).

Digital Object Identifier 10.1109/TSMCC.2006.876045

II. Objectives

Our aim is to develop a framework for experimenting on music style automatic recognition from symbolic representation of melodies (digital scores) by using shallow structural features, like melodic, harmonic, and rhythmic statistical descriptors. This framework involves all the usual stages in a pattern recognition system, like feature extraction, feature selection, and classification stages, in such a way that new features and corpora from different musical styles can be easily incorporated and tested.

Our working hypothesis is that melodies from a same musical genre may share some common features, permitting a suitable pattern recognition system, based on statistical descriptors, to assign the proper musical style to them.

Initially, two well-defined music styles, like jazz and classical, have been chosen as a workbench for our experiments. The initial results have been encouraging (see [15]), but the method performance for different classification algorithms, descriptor models, and parameter values needed to be thoroughly tested. This way a framework for musical style recognition can be set up, where new features and new musical styles can be easily incorporated and tested.

In this paper, we first present the proposed methodology, describing the musical data, the descriptors, and the classifiers that have been used. The initial set of descriptors are analyzed to test their contribution to the musical style separability. These procedures will permit us to build reduced models, discarding not useful descriptors. Then, the classification results obtained with each classifier, and their analysis with respect to the different description parameters, are presented. Finally, conclusions and possible lines of further work are discussed.

III. Methodology

In this section, we first present the music sources from which the experimental framework has been established. Second, the details of the statistical features from the musical data are described. Next, the feature selection procedure that led us to reduced models is explained. Then, the parameter space is discussed, and, finally, the classifier implementation and tuning are presented.

A. Musical Data

MIDI files from jazz and classical music were collected. These styles were chosen due to the general agreement in the musicology community about their definition and limits. Classical melody samples were taken from works by Mozart, Bach, Schubert, Chopin, Grieg, Vivaldi, Schumann, Brahms, Beethoven, Dvorak, Haendel, Paganini, and Mendelssohn. Jazz music samples were standard tunes from a variety of well-known jazz authors including Charlie Parker, Duke Ellington, Bill Evans, Miles Davis, etc. The MIDI files are composed of several tracks, one of them being the melody track from which the input data are extracted.[1] The corpus is made up of a total of 110 MIDI files, 45 of them being classical music and 65 being jazz music. The length of the corpus is around 10 000 bars (more than 6 h of music). Table I summarizes the distribution of bars from each style. This dataset is available for research purposes on request to the authors.

TABLE I
Distribution of Melody Length in Bars

	Min.	Max.	Avg.	Total	% of total
JAZZ	16	203	73	4734	47.5%
CLAS	44	297	116	5227	52.5%

This is a quite heterogeneous corpus, not specifically created to test our system but collected from different sources, ranging from websites to private collections without any processing before entering the system, except for manually checking for the presence and correctness of key, tempo, and meter meta-events as well as the presence of a monophonic melody track. The original conditions under which the MIDI files were created are unknown. They may be human-performed tracks or sequenced tracks (i.e., generated from scores) or even something of both worlds. Nevertheless, most of the MIDI files seem to fit a rather common scheme: a human-performed melody track with several sequenced accompaniment tracks.

The monophonic melodies consist of a sequence of musical events that can be either notes or silences. The pitch of each note can take a value from 0 to 127, encoded together with the MIDI note onset event. Each of these events at time t has a corresponding note off event at time $t + d$, (d being the note duration measured in ticks[2]). Time gaps between a note off event and the next note onset event are silences.

B. Description Scheme

A description scheme has been designed based on descriptive statistics that summarize the content of the melody in terms of pitches, intervals, durations, silences, harmonicity, rhythm, etc. This kind of statistical description of musical content is sometimes referred to as *shallow structure description* [16].

Each sample is a vector of musical descriptors computed from each melody segment available (see Section III-C for a discussion about how these segments are obtained). Each vector is labeled with the style of the melody to which the segment belongs to. We have defined an initial set of descriptors based on a number of feature categories that assess the melodic, harmonic, and rhythmic properties of a musical segment, respectively.

This initial model is made up of 28 descriptors summarized in Table II and described as follows.

1) Overall descriptors: *Number of notes, number of significant silences, and number of not significant silences*. The adjective *significant* stands for silences explicitly written in the underlying score of the melody. In MIDI files, short gaps between consecutive notes may appear due to interpretation nuances like *stacatto*. These gaps (interpretation silences) are not considered significant silences since they

[1] All the melodies are written in the 4/4 meter. Anyway, any other meter could be used because the measure structure is not used in any descriptor computation. All the melodies are monophonic sequences (at most one note is playing at any given time).

[2] A *tick* is the basic unit of time in a MIDI file and is defined by the resolution of the file, measured in ticks per beat.

TABLE II
MUSICAL DESCRIPTORS

Category	Descriptors
Overall	Number of notes Number of significant silences Number of non-significant silences
Pitch	Pitch range Average pitch Dev. pitch
Note duration	Note duration range Avg. note duration Dev. note duration
Silence duration	Silence duration range Avg. silence duration Dev. silence duration
Inter Onset Interval	IOI range Avg. IOI Dev. IOI
Pitch interval	Interval range Avg. interval Dev. interval
Non-diatonic notes	Num. non-diatonic notes Avg. non-diatonic degrees Dev. non-diatonic degrees
Syncopation	Number of syncopes
Normality	Pitch distrib. normality Note duration distrib. normality Silence duration distrib. normality IOI distrib. normality Interval distrib. normality Non-diatonic degree distrib. normality

should not appear in the score. To make a distinction between kinds of silences is not possible from the MIDI file, and it has been made based on the definition of a silence duration threshold. This value has been empirically set to a duration of a 16th note. All silences with longer or equal duration than this threshold are considered significant.

2) Pitch descriptors: *Pitch range* (the difference in semitones between the highest and the lowest note in the melody segment), *average pitch* relative to the lowest pitch, and *standard deviation of pitches* (provides information about how the notes are distributed in the score).
3) Note duration descriptors (measured in ticks and computed using a time resolution of $Q = 48$ ticks per bar[3]): *Range*, *average* (relative to the minimum duration), and *standard deviation* of note durations.
4) Significant silence duration descriptors (in ticks): *Range*, *average* (relative to the minimum), and *standard deviation*.
5) Interonset interval (IOI) descriptors (an IOI is the distance, in ticks, between the onsets of two consecutive notes[4]): *Range*, *average* (relative to the minimum), and *standard deviation*.
6) Interval descriptors (difference in absolute value between the pitches of two consecutive notes): *Range*, *average* (relative to the minimum), and *standard deviation.*
7) Harmonic descriptors:
   a) *Number of nondiatonic notes*. An indication of frequent excursions outside the song key (extracted from the MIDI file) or modulations.
   b) *Average degree of nondiatonic notes*. Describes the kind of excursions. This degree is a number between 0 and 4 that indexes the nondiatonic notes of the diatonic scale of the tune key, which can be major or minor key[5]
   c) *Standard deviation of degrees of nondiatonic notes*. Indicates a higher variety in the nondiatonic notes.
8) Rhythmic descriptor: *Number of syncopations*. Notes that do not begin at measure beats but in some places between them (usually in the middle) and that extend across beats.
9) Normality descriptors. They are computed using the D'Agostino statistic for assessing the distribution normality of the n values v_i in the segment for pitches, durations, intervals, etc. The test is performed using the following equation:

$$D = \frac{\sum_i (i - ((n+1)/(2))v_i}{\sqrt{n^3 \left(\sum_i v_i^2 - (1/n)\left(\sum_i v_i\right)^2\right)}}. \qquad (1)$$

The descriptors of this category computed for the analyzed segment are the normality values of the following:
   a) *pitch distribution;*
   b) *note duration distribution;*
   c) *IOI distribution;*
   d) *silence duration distribution;*
   e) *interval distribution;*
   f) *nondiatonic notes distribution*.

For pitch and interval properties, the range descriptors are computed as maximum minus minimum values, and the average-relative descriptors are computed as the average value minus the minimum value (only considering the notes in the segment). For durations (note duration, silence duration, and IOI descriptors), the range descriptors are computed as the ratio between the maximum and minimum values, and the average-relative descriptors are computed as the ratio between the average value and the minimum value.

This descriptive statistics is similar to histogram-based descriptions used by other authors [7], [8] that also try to model the distribution of musical events in a music fragment. Computing the range, mean, and standard deviation from the distribution of musical items like pitches, durations, intervals, IOIs, and nondiatonic notes, we reduce the number of features needed (each histogram may be made up of tens of features). Other authors have also used this sort of descriptors to classify music [6], [17], mainly focusing on pitches.

C. Free Parameter Space

Given a melody track, the statistical descriptors presented above are computed from equal-length segments, defining a window of size ω measures. Once the descriptors of a segment

[3]This is call quantization. $Q = 48$ means that when a bar is composed of four beats, each beat can be divided, at most, into 12 ticks.

[4]Two notes are considered consecutive even in the presence of a silence between them.

[5]Nondiatonic degrees are: 0: ♭II, 1: ♭III (♮III for minor key), 2: ♭V, 3: ♭VI, 4: ♭VII. The key is encoded at the beginning of the melody track. It has been manually checked for correctness in our data.

have been extracted, the window is shifted δ measures forward to obtain the next segment to be described. Given a melody with $m > 0$ measures, the number of segments s of size $\omega > 0$ obtained from that melody is

$$s = \begin{cases} 1, & \text{if } \omega \geq m \\ 1 + \left\lceil \frac{m-\omega}{\delta} \right\rceil, & \text{otherwise} \end{cases} \quad (2)$$

showing that at least one segment is extracted in any case (ω and s are positive integers; m and δ may be positive fractional numbers).

Taking ω and δ as free parameters in our methodology, different datasets of segments have been derived from a number of values for those parameters. The goal is to investigate how the combination of these parameters influences the segment classification results. The exploration space for this parameters will be referred to as $\omega\delta$-space. A point in this space is denoted as $\langle \omega, \delta \rangle$.

ω is the most important parameter in this framework since it determines the amount of information available for the descriptor computations. Small values for ω would produce windows containing few notes, providing little reliable statistical descriptors. Large values for ω would lead to merge, probably different, parts of a melody into a single window and they also produce datasets with fewer samples for training the classifiers [see (2)]. The value of δ would affect mainly the number of samples in a dataset. A small δ value combined with quite large values for ω may produce datasets with a large number of samples [see also (2)]. The details about the values used for these parameters can be found in Section IV.

D. Feature Selection Procedure

The features described above have been designed according to those used in musicological studies, but there is no theoretical support for their style classification capability. We have applied a selection procedure in order to keep those descriptors that better contribute to the classification. The method assumes feature independence, that is not true in general, but it tests the separability provided by each descriptor independently, and uses this separability to obtain a descriptor ranking.

Consider that the M descriptors are random variables $\{X_j\}_{j=1}^{M}$, whose N sample values are those of a dataset corresponding to a given $\omega\delta$-point. We drop the subindex j for clarity because all the discussion applies to each descriptor. We split the set of N values for each descriptor into two subsets: $\{X_{C,i}\}_{i=1}^{N_C}$ are the descriptor values for classical samples and $\{X_{J,i}\}_{i=1}^{N_J}$ are those for the jazz samples, where N_C and N_J are the number of classical and jazz samples, respectively. X_C and X_J are assumed to be independent random variables since both sets of values are computed from different sets of melodies. We want to know whether these random variables belong to the same distribution or not. We have considered that both sets of values hold normality conditions, and assuming that the variances for X_C and X_J are different in general, the test contrasts the null hypothesis $H_0 \equiv \mu_C = \mu_J$ against $H_1 \equiv \mu_C \neq \mu_J$. If H_1 is concluded, it is an indication that there is a clear separation between the values of this descriptor for the two classes and so it is a good feature for style classification. Otherwise, it does not seem to provide separability between the classes.

The following statistical for sample separation has been applied:

$$z = \frac{|\bar{X}_C - \bar{X}_J|}{\sqrt{s_C^2/N_C + s_J^2/N_J}} \quad (3)$$

where $\bar{X}_C$ and $\bar{X}_J$ are the means and s_C^2 and s_J^2 the variances for the descriptor values for both classes. The greater the z value is, the wider the separation between both sets of values is for that descriptor. A threshold to decide when H_0 is more likely than H_1, that is to say, the descriptor passes the test for the given dataset, must be established. This threshold, computed from a t-student distribution with infinite degrees of freedom and a 99.7% confidence interval, is $z = 2.97$. Furthermore, the z value permits to arrange the descriptors according to their separation ability.

When this test is performed on a number of different $\omega\delta$-point datasets, a threshold on the number of passed tests can be set as a criterion to select descriptors. This threshold is expressed as a minimum percentage of tests passed. Once the descriptors are selected, a second criterion for grouping them permits to build several descriptor models incrementally. First, selected descriptors are ranked according to their z value averaged over all tests. Second, descriptors with similar z values in the ranking are grouped together. This way, several descriptor groups are formed, and new descriptor models can be formed by incrementally combining these groups. See Section IV-A for the models that have been obtained.

E. Classifier Implementation and Tuning

Three algorithms from different classification paradigms have been used for style recognition. Two of them are fully supervised methods: the Bayesian classifier and the k-nearest neighbor (k-NN) classifier [18]. The other one is an unsupervised learning neural network, the SOM [19].

The Bayesian classifier is parametric and, when applied to a two-class problem, computes a discriminant function

$$g(X) = \log \frac{P(X \mid \omega_1)}{P(X \mid \omega_2)} + \log \frac{\pi_1}{\pi_2} \quad (4)$$

for a test sample X, where $P(X \mid \omega_i)$ is the conditional probability density function for class i and π_i are the priors of each class. Gaussian probability density functions for each style are assumed for each descriptor. Means and variances are estimated separately for each class from the training data. The classifier assigns a sample to ω_1 if $g(X) > 0$ and to ω_2 otherwise. The decision boundaries, where $g(X) = 0$, are in general hyperquadrics in the feature space.

The k-NN classifier uses an Euclidean metrics to compute the distance between the test sample and the training samples. The style label is assigned to the test sample by a majority decision among the nearest k training samples (the k-neighborhood).

SOM are neural methods that are able to obtain approximate projections of high-dimensional data distributions into low-dimensional spaces, usually bidimensional. Within the map,

TABLE III
SOM TRAINING PARAMETERS

training	iterations	neighb.rad.	learn.rate
coarse	3,000	12	0.1
fine	30,000	4	0.05

different clusters in the input data can be located. These clusters can be semantically labeled to characterize the training data and also hopefully future new inputs.

For SOM implementation and graphic representations, the SOM_PAK software [20] has been used. After some exploratory experiments, a 16×8 bidimensional map geometry has been eventually used. An hexagonal topology for unit connections and a bubble neighborhood have been selected for training. The radius of this neighborhood is equal for all the map units and decreases as a function of time. The training was done in two phases: a first fast coarse training and a second fine tuning phase (see Table III for the different training parameters). The metrics used to compute distances among samples is the Euclidean distance.

IV. EXPERIMENTS AND RESULTS

A. Feature Selection Results

The feature selection test presented in Section III-D has been applied to datasets corresponding to 100 randomly selected points of the $\omega\delta$-space. This is motivated by the fact that the descriptor computation is different for each ω and the set of values is different for each δ, and so the best descriptors may be different for different $\omega\delta$-points. Thus, by choosing a set of such points, the sensitivity of the classification to the feature selection procedure can be analyzed. Being a random set of points is a good tradeoff decision to minimize the risk of biasing this analysis.

The descriptors were sorted according to the average z value ($\bar{z}$) computed for the descriptors in the tests. The list of sorted descriptors is shown in Table IV . The $\bar{z}$ values for all the tests and the percentage of passed tests for each descriptor are displayed. In order to select descriptors, a threshold on the number of passed tests has been set to 95%. This way, those descriptors that failed the separability hypothesis in more than a 5% of the experiments were discarded from the reduced models. Only 12 descriptors out of 28 were selected. In the rightmost column, the reduced models in which the descriptors were included are presented. Each model is denoted with the number of descriptors included in it.

Three reduced size models have been chosen, with 6, 10, and 12 descriptors. This models are built according to the $\bar{z}$ value as displayed in Fig. 1. The biggest gaps in the $\bar{z}$ values for the sorted descriptors led us to group the descriptors in three reduced models. Note also that the values for $\bar{z}$ show a small deviation, showing that the descriptor separability is quite stable in the $\omega\delta$-space.

It is interesting to remark that at least one descriptor from each category of those defined in Section III-B were selected for a reduced model. The best represented categories were pitches and intervals, suggesting that the pitches of the notes and the relation among them are the most influent features for this problem. From the statistical point of view, standard deviations were the most important features, since five from six possible ones were selected.

TABLE IV
FEATURE SELECTION RESULTS

descriptor	$\bar{z}$	passed tests	models
Number of notes	22.5	100%	6,10,12
Average pitch	22.3	100%	6,10,12
Pitch range	22.2	100%	6,10,12
Interval range	20.3	100%	6,10,12
Syncopation	19.6	100%	6,10,12
Dev. pitch	18.7	100%	6,10,12
number of significant silences	14.2	100%	10,12
Interval distrib. normality	14.2	100%	10,12
Dev. interval	14.0	100%	10,12
Dev. IOI	13.2	97%	10,12
Dev. note duration	9.3	95%	12
Dev. non-diatonic degrees	9.1	100%	12
Dev. silence duration	6.3	94%	–
Silence duration range	6.1	87%	–
Note duration distrib. normality	6.0	89%	–
Avg. note duration	5.6	71%	–
Avg. silence duration	5.1	85%	–
Avg. non-diatonic degrees	4.9	66%	–
IOI range	4.7	53%	–
number of non-significant silences	4.5	76%	–
Silence duration distrib. normality	4.3	45%	–
Avg. IOI	4.2	53%	–
Non-diatonic degree distrib. normality	3.5	39%	–
Note duration range	3.3	34%	–
Pitch distrib. normality	2.6	25%	–
Num. non-diatonic notes	2.5	32%	–
IOI distrib. normality	2.2	20%	–
Avg. interval	1.7	14%	–

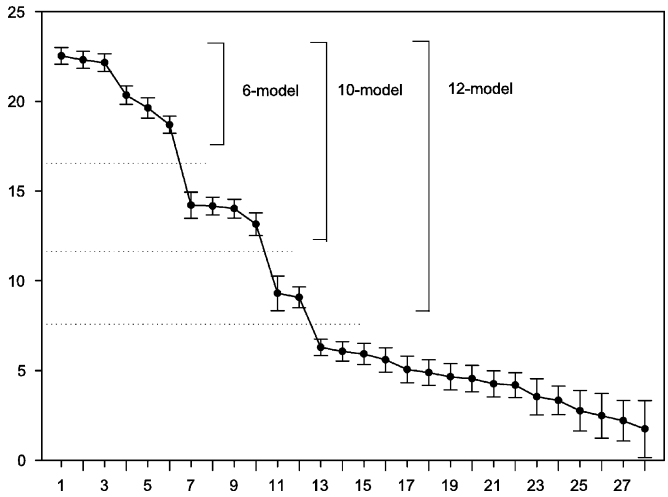

Fig. 1. Values for $\bar{z}$ for each descriptor as a function of their order numbers. The relative deviations for $\bar{z}$ in all the experiments are also displayed. The biggest gaps for $\bar{z}$ and the models are outlined.

B. $\omega\delta$-Space Framework

The melodic segment parameter space has been established as follows:

$$\omega = 1, \ldots, 100 \tag{5}$$

and for each ω

$$\delta = \begin{cases} 1, \ldots, \omega, & \text{if } \omega \leq 50 \\ 1, \ldots, 20, & \text{otherwise.} \end{cases} \quad (6)$$

The range for δ when $\omega > 50$ has been limited to 20 due to the very few number of samples obtained with large δ values for this ω range. This setup involves a total of 2275 points $\langle \omega, \delta \rangle$ in the $\omega\delta$-space. A number of experiments have been made for each of these points: one with each classifier (Bayes, NN, and SOM) for each of the four description models discussed in Section IV-A. Therefore, 12 different experiments for each $\omega\delta$-point have been made, denoted by $(\omega, \delta, \mu, \gamma)$, where $\mu \in \{6, 10, 12, 28\}$ is the description model and $\gamma \in \{\text{Bayes}, \text{NN}, \text{SOM}\}$ the classifier used.

In order to obtaining reliable results, a tenfold crossvalidation scheme has been carried out for each of the $(\omega, \delta, \mu, \gamma)$ experiments, making ten subexperiments with about 10% of samples saved for test in each subexperiment. The success rate for each $(\omega, \delta, \mu, \gamma)$ experiment is averaged for the ten subexperiments.

The partitions were made with the MIDI files to make sure that training and validation sets do not share segments from any common melody. Also the partitions were made in such a way that the relative number of measures for both styles were equal to those for the whole training set. This permits us to estimate the prior probabilities for both styles once and then use them for all the subexperiments. Once the partitions have been made, segments of ω measures are extracted from the melody tracks and labeled training and test datasets containing μ-dimensional descriptor vectors are constructed.

To summarize, 27 300 experiments consisting of ten subexperiments for each one, have been carried out. The maximum number of segments extracted is $s = 9339$ for the $\omega\delta$-point $\langle 3, 1 \rangle$. The maximum for s is not located at $\langle 1, 1 \rangle$ as expected because segments not containing at least two notes are discarded. The minimum is $s = 203$ for $\langle 100, 20 \rangle$. The average number of segments in the whole $\omega\delta$-space is 906. The average proportion of jazz segments is 36% of the total number of segments, with a standard deviation of about 4%. This is a consequence of the classical MIDI files having a greater length in average than jazz files, although there are less classical files than jazz files.

C. Classification Results

Each $(\omega, \delta, \mu, \gamma)$ experiment has an average success rate, obtained from the cross-validation scheme discussed in the previous section. The results presented here are based on those rates.

1) Bayes Classifier: For one subexperiment in a point in the $\omega\delta$-space, all the parameters needed to train the Bayesian classifier are estimated from the particular training set, except for the priors of each style, that are estimated from the whole set, as explained above.

Fig. 2 shows the classification results with the Bayesian classifier over the $\omega\delta$-space for the 12-descriptor model. This was one of the best combination of model and classifier (89.5%) in average for all the experiments. The best results were found around $\langle 58, 1 \rangle$, where a 93.2% average success was achieved.

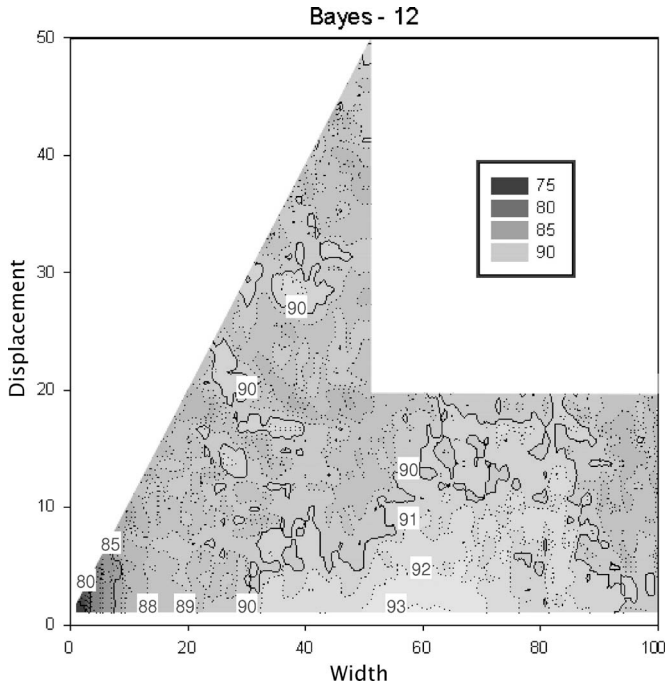

Fig. 2. Recognition percentage in the $\omega\delta$-space for the Bayesian classifier with the 12-descriptor model. Numbers on top of level curves indicate the recognition percentage at places on the curve. The best results (around 93.2%) are found in the lighter area, with large widths and small displacements.

The best results for style classification were expected to be found for moderate ω values, where enough musical events to calculate reliable statistical descriptors are contained in a segment, while musical events located in other parts of the melody are not mixed in a single segment. But the best results are generally obtained with a combination of large ω values and small δ. Experiments for $\omega = \infty$ (taking the whole melody as a single segment) are discussed in Section IV-C4.

The worst results occurred for small ω due to the few musical events at hand when extracting a statistical description for such a small segment, leading to nonreliable descriptors for the training samples.

All the three reduced models outperformed the 28-descriptor model (see Fig. 3 for a comparison between models for $\delta = 1$), except for $\omega \in [20, 30]$, where the 28-descriptor model obtains similar results for small values of δ. For some reason, the particular combination of ω and δ values in this range results in a distribution of descriptor values in the training sets that favors this classifier.

The overall best result (95.5% of average success) for the Bayesian classifier has been obtained with the ten-descriptor model in the point $\langle 98, 1 \rangle$. See Table V for a summary of best results (indices represent the $\langle \omega, \delta \rangle$ values for which the best success rates were obtained). About 5% of the subexperiments (4556 out of 91 000) for all models yielded a 100% classification success.

2) k-NN Classifier: Before performing the main experiments for this classifier, a study of the evolution of the classification as a function of k was designed in order to test the influence of this parameter in the classification task. The results

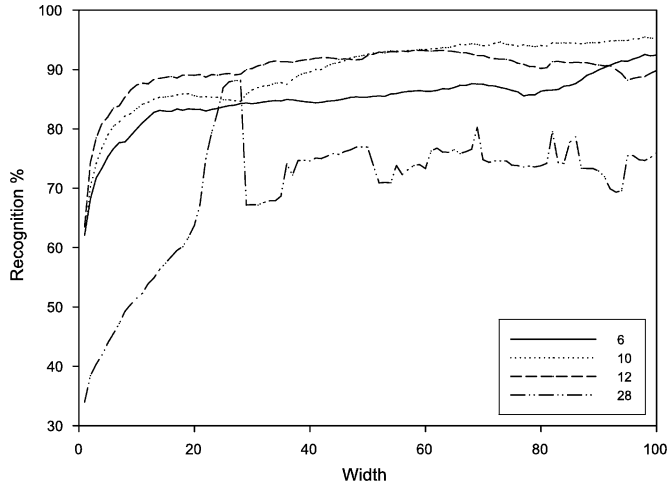

Fig. 3. Bayes recognition results for the different models versus the window width, with a fixed $\delta = 1$.

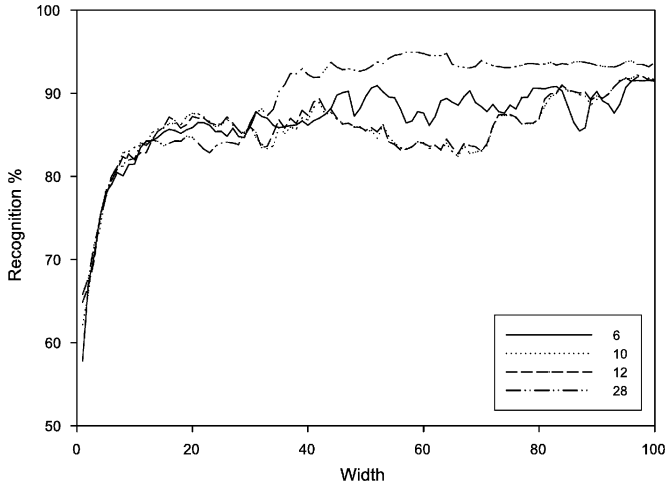

Fig. 5. NN recognition results for the different models versus the window width, with a fixed $\delta = 1$.

TABLE V
BEST SUCCESS RATES

model	Bayes	NN	SOM
6	$93.2_{\langle 100,2\rangle}$	$94.0_{\langle 91,16\rangle}$	$90.7_{\langle 19,4\rangle}$
10	$95.5_{\langle 98,1\rangle}$	$92.6_{\langle 99,19\rangle}$	$88.1_{\langle 20,2\rangle}$
12	$93.2_{\langle 58,1\rangle}$	$92.6_{\langle 98,19\rangle}$	$88.9_{\langle 23,6\rangle}$
28	$89.5_{\langle 41,33\rangle}$	$96.4_{\langle 95,13\rangle}$	$86.5_{\langle 23,5\rangle}$

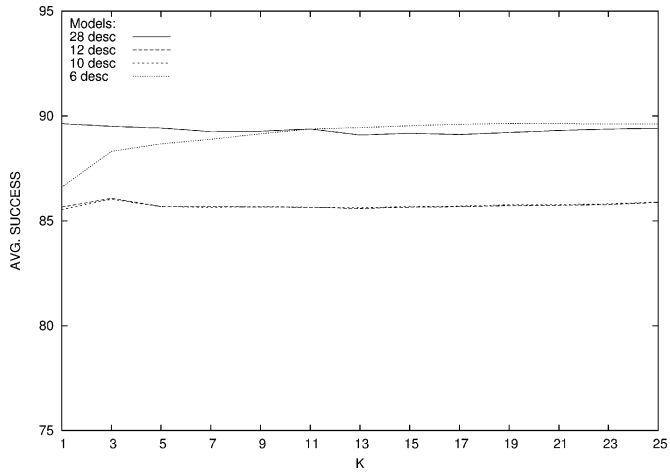

Fig. 4. Evolution of k-NN recognition for the different models against values of k.

are displayed in Fig. 4. Recognition percentage is averaged for all $\langle \omega, 1\rangle$ points. Note that there is almost no variation in the recognition rate as k increases, except a small improvement for the six-descriptor model. Thus, the simplest classifier ($k = 1$) was selected to avoid unnecessary time consumption due to the very large number of experiments to be performed.

Once the classifier has been set, the results for the different models were obtained and are displayed in Fig. 5 for $\delta = 1$. All models performed comparatively for $\omega \leq 35$. For $\omega > 35$, the 28-descriptor model begins to perform better than the reduced models. Its relatively high dimensionality and a greater dispersion in the samples (the larger the ω, the higher the probability of different musical parts to be contained in the same segment) causes larger distances among the samples, making the classification task easier for the k-NN.

The best results (96.4%) were obtained for the point $\langle 95, 13\rangle$ with the 28-descriptor model. The best results for all the models have been consistently obtained with very large segment lengths (see Table V). The percentage of perfect (100%) classification subexperiments amounts to 18.7% (17 060 out of 91 000).

For the whole $\omega\delta$-space, the NN classifier obtained an 89.2% in average with the 28-descriptor model, while the other models yielded similar rates, around 87%. The behavior of the 10- and 12-descriptor models was almost identical over the parameter space (Fig. 5) and for the different tested values for k (Fig. 4).

3) SOM Classifier: The SOMs were trained using the parameters shown in Table III and discussed in Section III-E. For each $\langle \omega, \delta\rangle$ subexperiment, at least three maps were initialized and trained before choosing that map with the minimum average quantisation error. The SOM was then labeled in a supervised way, using the training set as calibration set. Then the labeled map is used to classify test samples.

An example of a labeled map is shown in Fig. 6. Gray levels represent the distances between neighbor units, a darker gray level indicating a greater distance between units. Note that the labels for both styles trend to cluster in different parts of the map and how the calibration process has located the jazz labels mainly on the left zone and those corresponding to classical melodies on the right. Some units may be labeled with both music styles if they are activated by samples from both of them. In those cases, a single label is assigned to the unit according to the class that achieved the higher number of activations.

The general trend for all models [see Fig. 7(a)] was to give good classification results with $\omega \leq 20$ and small δ values. For larger segment lengths, the classification results worsen as ω increases, with large dispersion in the results, and no model seems to be better than the others. In any case, the six-descriptor model performed better on average (see Table VI) and provided the best success rate (90.7%) at $\langle 19, 4\rangle$.

The degradation of results for large ω is due to the fixed map dimensions used across the whole $\omega\delta$-space. For large segment

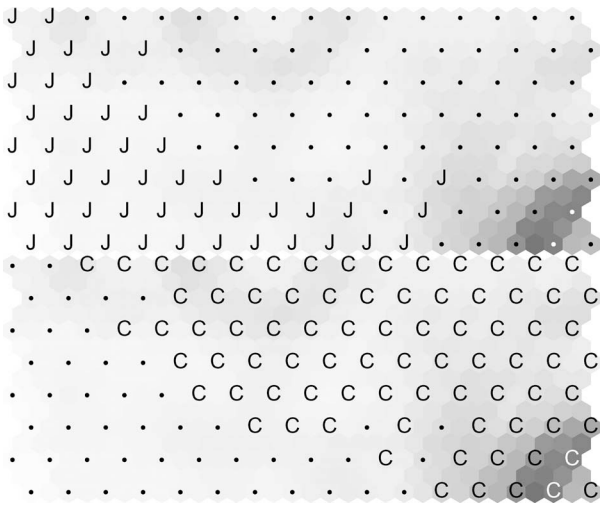

Fig. 6. SOM map labeled with (top) jazz and (bottom) classical for six-descriptor model, $\omega = 19$ and $\delta = 4$.

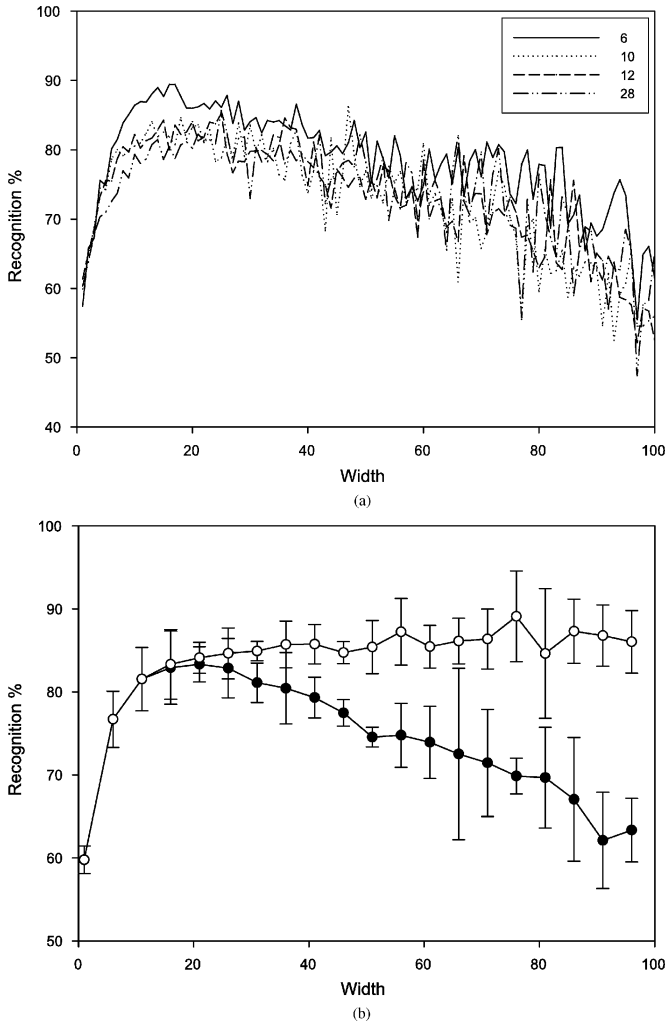

Fig. 7. (a) SOM recognition results for the different models against the window width, with a fixed $\delta = 1$. (b) The same results averaged for all models (–●–), and averaged only using the classified samples (–○–) (one point every five points is displayed for clarity).

TABLE VI
AVERAGES AND STANDARD DEVIATIONS OF SUCCESS RATES

model	Bayes	NN	SOM
6	84.2 ± 2.0	87.4 ± 2.9	77.5 ± 5.3
10	88.5 ± 3.2	86.9 ± 2.5	73.0 ± 6.3
12	89.5 ± 1.7	87.1 ± 2.5	73.0 ± 6.3
28	71.1 ± 6.3	89.2 ± 4.5	72.8 ± 5.4

TABLE VII
AVERAGE SUCCESS RATES FOR WHOLE MELODY SEGMENT LENGTH ($\omega = \infty$)

model	Bayesian	NN	SOM
6	88.0	87.0	55.0
10	91.0	88.0	54.0
12	91.0	88.0	52.0
28	79.0	93.0	56.0

lengths, the number of samples available for training seems to be not enough for a good coverage of the map, resulting in an excessive number of unlabeled units and, as a consequence, a high ratio of nonclassified test samples. When unclassified samples are not considered, this degradation does not occur [see Fig. 7(b)]. A method to estimate SOM size as a function of the number of training samples available for each $\langle \omega, \delta \rangle$ could be applied to improve these results.

4) Whole Melody Segment Classification: The good results obtained for large ω called our attention to the question of how good would be the results of classifying whole melodies, instead of fragments, as presented so far. The first problem is the small number of samples available this way (110 samples for training and test). This is particularly hard for training the SOM. The results of these experiments are displayed in Table VII. The same ten-fold cross-validation scheme described in Section IV-B was used here. The results are comparable or even better than the average in the $\omega\delta$-space for Bayesian and NN classifiers, but SOMs were unable to perform well due to the small size of the training set.

In spite of this good behavior for Bayes and k-NN, this approach has a number of disadvantages. Training is always more difficult due to the smaller number of samples. The classification cannot be performed *online* in a real-time system because all the piece is needed in order to take the decision. There are also improvements to the presented methodology, like cooperative decisions using different segment classifications that cannot be applied to the complete melody approach.

5) Results Comparison: Bayesian and NN classifier performed comparatively and better than SOM. There were, in general, lower differences in average recognition percentages between models for NN than those found with the Bayesian classifier (see Table VI) probably due to its nonparametric nature.

An ANOVA test with Bonferroni procedure for multiple comparison statistics [21] was used to determine which combination of model and classifier gave the best classification results in average. According to this test, with the number of experiments performed, the required difference between any two recognition rates in Table VI must be at least 0.45123 in order to be considered statistically different at the 95% confidence level. Thus, it can be stated that Bayes classifier with 12-descriptor model

and NN classifier with 28-descriptor model perform comparatively well, and both outperform the rest of classifier and model combinations. The Bayes classifier has the advantage of using a reduced size description model.

In a recent work using the same dataset [22], several text categorization algorithms have been used to perform style recognition from whole melodies. In particular, a naive Bayes classifier with several multivariate Bernoulli and multinomial models are applied to binary vectors indicating the presence or absence of n-length words (sequences of n notes) in a melody. The work reported around 93% of success as the best performance. This is roughly the same best result reported here for the whole melody, although it is outperformed by the window classifications.

Results for the $\omega\delta$-space are hardly comparable with results by other authors because we used segments instead of complete melodies and because of the different datasets put under study by different authors. Nevertheless, a comparison attempt can be made with the results found in [6] for pairwise genre classification. The authors use information from all the tracks on the MIDI files except tracks playing on the percussion channel. In [16], a 94% accuracy for Irish folk music and jazz identification is reported as the best result. Unfortunately, they did not use classical music samples. This accuracy percentage is similar to our results with whole melody length segments and the NN classifier (93%). A study on the classification accuracy as a function of the input data length is also reported, showing a behavior similar to the one reported here: classification accuracy using statistical information reaches its maximum for larger segment lengths, as they reported a maximum accuracy for five classes with 4-min segment length. Our best results were obtained for $\omega > 90$ (see Table V).

V. Conclusion and Future Work

Our main goal in this work has been to test the capability of melodic, harmonic, and rhythmic statistical descriptors to perform musical style recognition. We have developed a framework for feature extraction, selection, and classification experiments, where new corpora, description models, and classifiers can be easily incorporated and tested.

We have shown the ability of three classifiers, based on different paradigms, to map symbolic representations of melodic segments into a set of musical styles. Jazz and classical music have been used as an initial benchmark to test this ability. The experiments have been carried out over a parameter space defined by the size of segments extracted from melody tracks of MIDI files and the displacement between segments sequentially extracted from the same source. A total of 273 000 classification subexperiments have been performed.

From the feature selection stage, a number of interesting conclusions can be drawn. From the musical point of view, pitches and intervals have been shown to be the most discriminant features. Other important features have been the number of notes and the rhythm syncopation. Although the former set of descriptors may be probably important in other style classification problems, probably these latter two have found their importance in this particular problem of classical versus jazz. From the statistical point of view, standard deviations were very relevant, since five from six possible ones were selected.

The general behavior for all the models and classifiers against the values for ω was to have bad classification percentages (around 60%) for $\omega = 1$, rapidly increasing to an 80% for $\omega \approx 10$ and then keep stable around a 90% for $\omega > 30$. This general trend supports the importance of describing large melody segments to obtain good classification results. The preferred values for δ were small, because they provide a higher number of training data.

Bayes and NN performed comparatively. The parametric approach preferred the reduced models but NN performed well with all models. In particular, with the complete model, without feature selection, it achieved very good rates, probably favored by the large distances among prototypes obtained with such a high dimensionality. The best average recognition rate has been found with the Bayesian classifier and the 12-descriptor model (89.5%), although the best result was obtained with the NN, which reached a 96.4% with $\omega = 95$ and $\delta = 13$.

The SOM classifier achieved results comparable to the other classifiers for $\omega < 20$, but they got worse for larger ω values, because of the number of unclassified samples and because a fixed map size was used for all the experiments (see Section IV-C3).

Also, whole melody classification experiments were carried out, removing the segment extraction and segment classification stage. This approach is simpler, faster, and provide comparative results even with few training samples, but has a number of disadvantages. It does not permit the use of online implementations where the system can input data and take decisions in real-time, since all the piece needs to be entered to the classifier in a single step. In addition, the segment classification approach permits to analyze a long theme by sections, performing local classifications.

An extension to this framework is under development, where a voting scheme for segments is used to collaborate in the classification of the whole melody. The framework permits the training of a large number of classifiers that, combined in a multiclassifier system, could produce even better results.

In the future, we plan to make use of all this methodology to test other kind of classifiers, like feedforward neural nets or support vector machines, and to explore the performance with a number of different styles.

Acknowledgment

The authors would like to thank C. Pérez-Sancho, F. Moreno-Seco, and J. Calera for their help, advise, and programming. Without their help this paper would have been much more difficult to finish. The authors also wish to thank the anonymous reviewers for their valuable comments.

References

[1] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," in *Proc. 4th ISMIR*, 2003, pp. 201–208.

[2] B. Whitman, G. Flake, and S. Lawrence, "Artist detection in music with minnowmatch," in *Proc. IEEE Workshop Neural Networks Signal Process.*, 2001, pp. 559–568.

[3] H. Soltau, T. Schultz, M. Westphal, and A. Waibel, "Recognition of music types," in *Proc. IEEE ICASSP*, 1998, pp. 1137–1140.
[4] F. Pachet, G. Westermann, and D. Laigre, "Musical datamining for EMd," presented at the Wedelmusic Conf., Florence, Italy, 2001.
[5] R. Dannenberg, B. Thom, and D. Watson, "A machine learning approach to musical style recognition," in *Proc. ICMC*, 1997, pp. 344–347.
[6] G. Tzanetakis, A. Ermolinskyi, and P. Cook, "Pitch histograms in audio and symbolic music information retrieval," *J. New Music Res.*, vol. 32, no. 2, pp. 143–152, Jun. 2003.
[7] B. Thom, "Unsupervised learning and interactive jazz/blues improvisation," in *Proc. AAAI*, 2000, pp. 652–657.
[8] P. Toiviainen and T. Eerola, "Method for comparative analysis of folk music based on musical feature extraction and neural networks," in *3rd Intl. Conf. Cognitive Musicol.*, 2001, pp. 41–45.
[9] W. Chai and B. Vercoe, "Folk music classification using hidden Markov models," presented at the Int. Conf. Artificial Intelligence, Las Vegas, NV, 2001.
[10] S. Dubnov and G. Assayag, *Mathematics and Music*. New York: Springer, 2002, ch. 9, pp. 147–158.
[11] E. Stamatatos and G. Widmer, "Music performer recognition using an ensemble of simple classifiers," in *Proc. ECAI*, 2002, pp. 335–339.
[12] P. van Kranenburg and E. Backer, "Musical style recognition—A quantitative approach," in *Proc. CIM*, 2004, pp. 106–107.
[13] P. P. Cruz-Alcázar, E. Vidal, and J. C. Pèrez-Cortes, "Musical style identification using grammatical inference: The encoding problem," in *Proc. CIARP*, 2003, pp. 375–382.
[14] G. Buzzanca, "A supervised learning approach to musical style recognition," presented at the 2nd Int. Conf. Music and Artificial Intelligence, Edinburgh, U.K., 2002.
[15] P. J. Ponce de León and J. M. Iñesta, "Musical style classification from symbolic data: A two style case study," in *Proc. 1st Computer Music Modeling and Retrieval Conference*, Lecture Notes in Computer Science, vol. 27771. Berlin, Germany: Springer, 2004, pp. 166–177.
[16] J. Pickens, "A survey of feature selection techniques for music information retrieval," Cent. Intell. Inf. Retr., Dept. Comput. Sci., Univ. Massachussetts, Amherst, Tech. Rep., 2001.
[17] S. G. Blackburn, "Content based retrieval and navigation of music using melodic pitch contours" Ph.D. dissertation, Dept. Electron. Comput. Sci., Univ. Southampton, Southampton, U.K., 2000.
[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, Wiley, 2000.
[19] T. Kohonen, "Self-organizing maps," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.
[20] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. (1995, Apr.). Som pak, the self-organizing map program package, v:3.1. SOM PAK 3.1, Lab. Comput. Inf. Sci. Univ. Technol., Helsinki, Finland. [Online] Available: http://www.cis.hut.fi/ research/som pak
[21] G. R. Hancock and A. J. Klockars, "The quest for alpha; developments in multiple comparison procedures in the quarter century since games (1971)," *Rev. Education. Res.*, vol. 66, no. 3, pp. 269–306, 1996.
[22] C. Peŕez-Sancho, J. M. Iñesta, and J. Calera-Rubio, "Style recognition through statistical event models," in *Proc. SMC*, 2004, pp. 135–140.

Pedro J. Ponce de León received the B.Sc. and M.Sc. degrees in computer science from the University of Alicante, Alicante, Spain, in 1997 and 2001, respectively, where he is currently working toward the Ph.D. degree.

Since 2002, he has been an Assistant Lecturer at the University of Alicante. His main research interests include machine learning, music information retrieval, and music perception and modeling, having published papers on this topics in several international journals and conference proceedings.

José M. Iñesta received the B.Sc. and Ph.D. degrees in physics from the University of Valencia, Valencia, Spain, in 1987 and 1994, respectively.

He was an Assistant Lecturer in the Jaume I University of Castellón. In 1998, he joined the University of Alicante, Alicante, Spain, where he is currently a Professor. He is the author or editor of seven books, 24 international journals, 37 book chapters, and more than 50 papers presented at international and national conferences. He has been involved in 21 research projects (national and international) covering medical applications of pattern recognition and artificial intelligence, robotics, or digital libraries, among other lines of work. His main research interests range now from image analysis to pattern recognition algorithms with applications in medicine, robotics, and computer music.