Multiobjective time series matching for audio classification and retrieval

Philippe Esling, Carlos Agon

***Abstract*—Seeking sound samples in a massive database can be a tedious and time consuming task. Even when metadata are available, query results may remain far from the timbre expected by users. This problem stems from the nature of query specification, which does not account for the underlying complexity of audio data. The *Query By Example* (QBE) paradigm tries to tackle this shortcoming by finding audio clips similar to a given sound example. However, it requires users to have a well-formed soundfile of what they seek, which is not always a valid assumption. Furthermore, most audio-retrieval systems rely on a single measure of similarity, which is unlikely to convey the perceptual similarity of audio signals. We address in this paper an innovative way of querying generic audio databases by *simultaneously* optimizing the temporal evolution of multiple spectral properties. We show how this problem can be cast into a new approach merging multiobjective optimization and time series matching, called *MultiObjective Time Series* (MOTS) matching. We formally state this problem and report an efficient implementation. This approach introduces a multidimensional assessment of similarity in audio matching. This allows to cope with the multidimensional nature of timbre perception and also to obtain a *set* of *efficient propositions* rather than a single *best* solution. To demonstrate the performances of our approach, we show its efficiency in audio classification tasks. By introducing a selection criterion based on the *hypervolume* dominated by a class, we show that our approach outstands the state-of-art methods in audio classification even with a few number of features. We demonstrate its robustness to several classes of audio distortions. Finally, we introduce two innovative applications of our method for sound querying.**

I. Introduction

The past decade has witnessed a growing interest in *content-based retrieval* for multimedia databases [81]. Large amount of work has been devoted to performing similarity queries over musical songs databases [15]. An intuitive way of finding songs has been shown to be *Query By Humming* (QBH) [85], which is now a popular content-based musical retrieval method. This paradigm allows finding a song in a large collection simply by humming its melody. Tracing back to the seminal work of Ghias et al. [27], QBH systems typically rely on *symbolic* representations of melodies, rather than audio databases. Sound sample databases induce a greater challenge, as they are more massive and grow faster than musical databases. Furthermore, sound samples do not benefit from the same high-level symbolic information that can be extracted from melodies. Therefore, such sets may require an overwhelming amount of time to find a particular sample. The *Query By Example* (QBE) paradigm tries to tackle this problem by finding audio clips similar to a given sound example based on their spectral properties. The first QBE system was proposed by Wold et. al [79] where sounds were represented by a vector of spectral features, which and compared with the Euclidean distance. This approach has subsequently been extended using larger sets of features [76] or other spectral transforms like the Discrete Cosine Transform (DCT) [70] and wavelet transform [43]. Several learning schemes have also been investigated like Nearest Feature Line (NFL) [44], Support Vector Machine (SVM) [31] or Gaussian Mixture Model (GMM) [35]. Other studies have focused on the temporal modeling of sounds, either by using templates of temporal energy [8] or Hidden Markov Model (HMM) [82]. Another stream of audio querying is *Semantic Audio Retrieval* [69] which tries to discover the relationship between semantic and acoustic spaces. This enables queries on semantic concepts rather than acoustic features. This approach was implemented with a mixture of experts [68] and extended with polysemy handling [10].

Generic audio retrieval is facing several problems that can be outlined from previous works in this field. First of all, metadata information is clearly insufficient to provide complex interactions. It seems difficult to maintain consistent and expressive metadata on large datasets. Semantic retrieval provides a way to avoid manual annotation but still requires an extensively annotated starting set. Furthermore, it is limited to descriptive facts and sounds clearly related to a production source. Most of the timbre 'qualities' cannot be captured using semantic concepts without subjective interpretation of data [21], [57]. This imposes severe limitations on the range of possible queries, restricted to a predetermined set of semantic classes. In order to avoid this drawback, the use of onomatopoeias have been proposed to provide a retrieval scheme for abstract sounds [72]. However, the choice of labels is still subjective and can result in conflicts on a semantic level with acoustically different sounds mapped to the same onomatopoeia. Some QBE systems use clustering before retrieval given that search time could be reduced by comparing the query only to a relevant cluster [34], [83]. However, building hierarchical classes implies that the database is created according to a specific dataset. Therefore, once the database is built, it loses flexibility and users have to adapt to this original hierarchy. Regarding timbre, several authors pointed out that it is a multidimensional phenomenon [53], [78], and psychoacoustic studies often use multidimensional spaces to classify sounds [29], [64]. Authors in the Music Information Retrieval (MIR) community have also pointed out the multifaceted nature of audio perception [21] and

that a single measure is unlikely to convey the perceptual similarity of audio signals [75]. Sound retrieval systems should be flexible enough so that depending on listeners and target timbres, variable influence could be put on different sound properties during similarity evaluations [51], but yet no current audio-retrieval system seems to address these limitations.

In this paper, we propose a new paradigm to tackle previously cited problems. Our system relates to [57] where sounds with or without a known cause are described by looking specifically at the temporal evolution of their acoustical properties. Sound clips are considered as short-duration *units of musical creativity* [14]. In order to provide flexibility in the database, we avoid the clustering paradigm by deliberately not interpreting data. Therefore, no assumptions are made on spectrum types and sounds can be of any nature. In order to provide more comprehensive and universal query conditions, we do not use semantic annotation and focus on the temporal evolution of timbre properties. Based on previous observations, we believe that the multidimensional nature of timbre perception should be taken into account in audio matching processes. Therefore, our system optimizes various spectral dimensions jointly, without mixing them into a single distance measure. We propose a new approach inspired by multiobjective optimization and time series matching, namely *MultiObjective Time Series* (*MOTS*) matching, which has never been addressed to our best knowledge. This framework allows to provide a multidimensional assessment of similarity in audio matching. By introducing a new classification criterion based on *hypervolume domination*, we show that our assumptions on required properties for audio-retrieval systems allow to outperform the state-of-art methods in classification tasks. We also show the robustness of our approach to different classes of audio distortions.

Finally, we consider a core problem of audio retrieval that lies in the query specification itself. As put forward by Donwie [21], audio queries are themselves complex and multifaceted musical information. Several authors pointed out that most users have only a vague idea of what they seek at the onset [42], [80]. Hence, they might search for aspects of the audio query but not exactly the same content. We show how MOTS query results handle this aspect by being presented to users in an informative way. Finally, when an example is unknown or difficult to generate, the query should help the user determine what he is seeking by being specified in a way as close as possible to the underlying nature of audio properties [59]. We present two potential applications of the MOTS approach for innovative audio querying in order to cope with the multidimensionality of timbre perception. First, the *MultiObjective Spectral Evolution Query (MOSEQ)* provides a flexible query specification by directly allowing users to draw schematic temporal shapes required for spectral features. Therefore, it bypasses the need for a specific example. Based on this paradigm, we introduce the *Query by Vocal Imitation (QVI),* which allows users to perform vocal imitations of desired properties. In both cases, the system displays the samples on a multidimensional space depending on how well they match the different dimensions.

The rest of this paper is organized as follows. We begin by reviewing existing approaches for content-based audio retrieval (Section II-A), time series analysis (Section II-B) and multiobjective optimization (Section II-C). We formally state the generic MOTS problem, its uniqueness towards existing researches (Section III) and describe efficient algorithms to handle it (Section III-C). We analyze the computational efficiency of these algorithms on massive datasets (Section III-D). As we also evaluate multidimensional similarity for classification tasks, we introduce two novel class selection criteria (Section IV-A). We analyze the accuracy of this classification approach, its robustness to various audio distortions and compare it to state-of-art results (Section IV-C). We present queries representations and introduce two potential applications for sound querying (Section V).

II. State of the arts

A. Content-based audio retrieval

Content-based audio retrieval has become a popular research field, notably through the appearance of QBH introduced by Ghias et al. [27]. Most of researches devoted to this subject are based on symbolic song databases and, therefore, use the notion of *pitch contour* [74], which is the sequence of relative differences in pitch between successive notes. Recently there has been studies that match songs directly from audio using the *melody slope* [49], [84], which is the continuous equivalent of the pitch contour. The matching process must be flexible enough to allow errors in the user's query and several approaches have been proposed such as HMMs [37], dynamic programming [55] or time series matching [86].

The QBE paradigm has been proposed in order to retrieve generic audio signals. QBE is based on the idea that users could find samples similar to a given example based on its spectral properties. The first QBE system was proposed by Wold et. al [79] where sounds were represented by a vector of mean, variance and autocorrelation values of spectral features. These vectors were then compared with the Euclidean distance as similarity metric. This approach known as *Bag-Of-Features* (BOF) has been extended using larger sets of features [76] or adding *relevance feedback* [59] in which the user selects its preferred results for refinement. Subramanya et al. [70] used frequency coefficients from spectral decompositions and showed the superiority of DCT. They later used the multiresolution property of the wavelet transform [71] and showed its robustness to noise. However, the selection of coefficients yields extremely large vectors, which may be unsustainable for massive datasets. This approach was extended in [43] by using multiple statistical values over wavelet coefficients. This allows hierarchical indexing, as proposed in [45] with a pyramidal algorithm which provides acceleration over previous approaches. Several indexing and learning schemes have also been investigated. Li [44] proposed

the Nearest Feature Line (NFL) based on the idea that, in feature space, lines between similar audio clips represent continuous deformations between class properties. Therefore, comparisons with queries are made with these feature lines. However, computing NFLs between every sound samples seems to induce a large computing and storage overhead. Other machine learning techniques like Boosting [32] or GMM [35] were studied, but they seem to be outperformed by the SVM-based approach.

Regarding temporal models, Cai et. al [8] proposed to use templates of temporal patterns for energy, harmonicity and pitch contour. Although they showed to improve the accuracy, this approach seems hardly scalable because of the relative simplicity of the patterns used. More comprehensive temporal modelization with HMMs [82] has been proposed, where comparison of HMM likelihoods with the query allows to obtain a ranked list of results. Casey [11] proposed to use the MPEG-7 feature set with an Independant Subspace Analysis (ISA) to obtain the most salient features of a sound. He further introduced a minimum entropy method [12] to train the HMM classifier which appears to outperform classical training. However, the ISA usually yields large computational overheads. The superiority of HMM cross-likelihood ratio has been shown over GMM [75] and feature histograms [33] for class-based QBE. However, these studies exhibited that all the approaches are highly sensitive to noise and low-quality sounds.

Another stream of generic audio querying is *Semantic Audio Retrieval* [69], which tries to discover the relationship between semantic and acoustic spaces in order to perform queries on semantic concepts rather than acoustic features. The idea is to model the semantic space as a multinomial model and use a probabilistic model to associate the related acoustic properties. This approach was implemented with a mixture of probability experts in [68]. Cano et. al [10] proposed to expand this approach with a taxonomy to avoid tag confusion and polysemy. They further used a NN classifier [9] with sounds linked to concepts. Barrington et. al [5] proposed a mapping in which each dimension indicates the relative importance of its semantic concept. Casey [13] proposed to use the Passive-Aggressive Model for Image Retrival (PAMIR) to establish the mapping between spaces. This method performs equivalently as GMM and SVM but seems to be faster. Semantic retrieval allows to circumvent the problem of manual annotation but still requires a starting set of annotated sounds. This also poses severe limitations for generic sounds properties which cannot be described objectively like synthesis sounds.

B. Time series analysis

A time series is a collection of values obtained from sequential measurements over time. It can be defined as an ordered sequence of n real-valued variables

$$T = (t_1, \ldots, t_n), t_i \in \mathbb{R} \tag{1}$$

A time series is often related to an underlying process observed at uniformly spaced *time instants* at a given *sampling rate*. Time series analysis originates from the desire to mimic our natural ability to visualize the *shape* of data. Indeed, instead of being trapped by small fluctuations we are able to abstract a notion of shape and spot similarities between patterns on various time scales almost instantly. Major time series tasks include query by content [24], anomaly detection [77], motif discovery [47] and classification [3]. This research field must handle the high dimensionality induced by working on time series data. Therefore, databases usually contain simplified *representations* $\bar{T}_s$ of the series T_s. which are models of reduced dimensionality such that $\bar{T}_s$ retains the essential characteristics of T_s. Another difficulty lies in defining a *similarity measure* $\mathcal{D}(T, U)$ between time series. This distance should allow recognition of perceptually similar *shapes* even though they are not mathematically identical. Finally, algorithms must scale to evergrowing massive datasets by using *indexing* techniques, which should provide minimal space consumption and computational complexity.

Query by content is the most active area of research in time series analysis. It is based on retrieving a set of solutions that are most similar to a query provided by the user.

Definition 1. *Query by content*. Given a query time series $Q = (q_1, ..., q_n)$ and a similarity measure $\mathcal{D}(Q, T)$, find the ordered list $\mathcal{L} = \{T_1, \ldots, T_m\}$ of time series in database DB, such that $\forall T_k, T_j \in \mathcal{L},\ k > j \Leftrightarrow \mathcal{D}(Q, T_k) \geq \mathcal{D}(Q, T_j)$.

Two types of queries are usually available. It is possible to specify a threshold ϵ and retrieve all series in the database whose similarity $\mathcal{D}(Q, T)$ with the query is less than ϵ (ϵ-*range query*). Obviously, the choice of this threshold is tedious and highly data-dependent. Alternatively, users can retrieve a set of solutions by constraining the number of series it should contain, ie. querying the K most similar series in the database ($K-$*Nearest Neighbors query*).

Time series analysis was at first devoted to this task, tracing back to the seminal work of Agrawal et al. [1] where the representation was based on coefficients obtained from a Discrete Fourier Transform (DFT). These coefficients were then indexed with an R*-tree [6]. False hits were then removed by using the Euclidean distance on complete time series. This paper laid the foundations of a reference framework which has later been extended by using properties of the DFT [61] or similar decompositions like Discrete Wavelet Transform (DWT) [16]. Several numerical transformations like Piecewise Linear Approximation (PLA) [66] and Piecewise Approximate Aggregation (PAA) [38] or symbolic representations like shape alphabets [2] and bit level approximations [62] have been proposed. For distance measures, numerous authors pointed out pitfalls when using L_p norms [20], [39]. Therefore, several proposals have been made to provide a similarity consistent with human intuition. Shape-based distances allow non-uniform comparison along the timeline like Dynamic Time Warping (DTW) [40] or Optimal Subsequence Bijection (OSB) [41]. Edit-based

distances like the Longest Common SubSequence (LCSS) [18] handle outliers by allowing gaps in the series. As a complete review of time series research is beyond the scope of this paper, we refer interested readers to [23].

In our study, we use the Symbolic Aggregate approXimation (SAX) [46] representation. The motivation behind this choice is two-fold. First, regarding accuracy and efficiency, SAX has been shown to consistently outperform other representations [20]. Second, this representation provides an efficient way to store temporal information and allows using the iSAX [67] indexing technique which has been devised to handle queries over million-sized databases. Finally, for computing similarity, we use the LB_Keogh [40] distance which enables efficient DTW computation. Allowing such non-linear distortions of the time axis is particularly relevant in audio applications and the effectiveness of these approaches has been shown in the field of QBH [86].

C. Multiobjective optimization

Multiobjective approaches were designed to handle problems where several objectives are required to be optimized simultaneously. Given a search space (also called *decision space*) S and a set of functions $F = \{f_1, ..., f_N\}$ to minimize over S, a multiobjective problem is defined by

$$\begin{cases} min & F(x) = \{f_1(x), ..., f_N(x)\} \\ s.t. & x \in S \end{cases} \quad (2)$$

Given this problem, the space defined as

$$C = \{(f_1(x), ..., f_N(x)) \mid x \in S\} \quad (3)$$

is called the *criteria space*. Usually, the ideal solution x^*, which is the global minimum for all criteria, does not exist. Therefore, multiobjective problems cannot be solved with a single "perfect" solution, but rather with a *set* of *efficient solutions*. In order to obtain this collection of tradeoffs among objectives, a relaxed notion of optimality needs to be adopted. This concept of dominance was introduced by Edgeworth [22] and later generalized by Pareto [54] and is called the *Pareto optimum*. A Pareto solution is optimal in each direction of optimization as it is not dominated in *every* objective. Hence, it is impossible to find another solution that improves the *complete* set of criteria of a Pareto solution.

Definition 2. Let x and y be two points of a search space S. We say that a solution y is *Pareto dominated* by a solution x (noted $x \preceq y$) iff it is dominated in every dimension

$$\forall n \in \{1, ..., N\}, \ f_n(x) \leq f_n(y) \quad (4)$$

We say that x strictly dominates y (noted $x \prec y$) iff $\exists n_0 \in \{1, ..., N\}$ such that $f_{n_0}(x) < f_{n_0}(y)$.

The dominance relation $\prec$ induces only a partial order on the criteria space, as shown in Figure 1. For any element x, the criteria space is divided into three regions depending on the dominance relation between x and the corresponding subspaces. We call these subspaces $\mathcal{S}_\prec$, $\mathcal{S}_\succ$ and $\mathcal{S}_?$. $\mathcal{S}_\prec$ contains the elements that dominates x ($\forall y \in \mathcal{S}_\prec, y \prec x$). $\mathcal{S}_\succ$ is the subspace whose elements are dominated by x ($\forall y \in \mathcal{S}_\succ, x \prec y$). Finally, the elements of $\mathcal{S}_?$ simply cannot be compared to x as they are not dominated nor dominate x ($\forall y \in \mathcal{S}_?, x \not\preceq y \wedge y \not\preceq x$). The distribution of non-dominated elements of a decision space is called the *Pareto front*. Solving a multiobjective problem can, therefore, be summarized as the discovery of the Pareto front. Figure 2 depicts a search space in the bi-objective case, where this front emerges from non-dominated solutions. From a strict point of view, none of the Pareto solutions can be preferred to others.

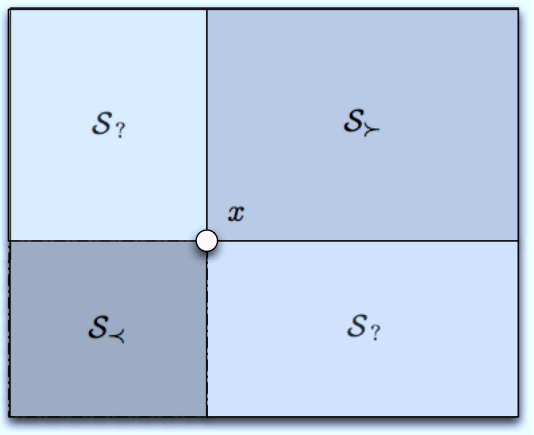

Figure 1. *Pareto dominance* relations in a bi-criteria space. Any point x of the criteria space divides it into three sub-spaces depending on the dominance relation. $\mathcal{S}_\prec$ contains the elements that dominates x ($\forall y \in \mathcal{S}_\prec, y \prec x$). Elements of $\mathcal{S}_\succ$ are dominated by x ($\forall y \in \mathcal{S}_\succ, x \prec y$). Finally, the elements of $\mathcal{S}_?$ simply cannot be compared to x as they are not dominated nor dominate x ($\forall y \in \mathcal{S}_?, x \not\preceq y \wedge y \not\preceq x$).

III. MultiObjective Time Series (MOTS) Matching

Now equipped with the core notions of multiobjective optimization and time series matching, we introduce the generic *MOTS* matching problem. We demonstrate the novelty and complexity of this problem and its relevance to find more flexible sets of solutions by allowing every direction of optimization.

A. Multiobjective time series matching

Given the previous definitions, a *MOTS* matching problem is defined as finding the efficient elements of a database that jointly minimize a set of time series distances

$$\begin{cases} min\ \mathcal{D}_Q^k(\mathcal{S}) & k \in \{1, ..., K\} \\ s.t.\ \mathcal{S} \in DB \end{cases} \quad (5)$$

with Q the query represented by a set of K time series and $\mathcal{S}$ an element of the database DB which contains time series corresponding to the same objectives as the query. Finally, $\mathcal{D}_Q^k(\mathcal{S})$ is the similarity between the k^{th} feature represented by time series Q_k and $\mathcal{S}_k$, i.e. $\mathcal{D}_Q^k(\mathcal{S}) = \mathcal{D}(Q_k, \mathcal{S}_k)$. We can already see here that part of the computational complexity of this problem arises from objective functions $\mathcal{D}_Q^k(\mathcal{S})$, which represent time series distances. As explained in Section II-B, the concept of time series similarity is remarkably subtle and implies a

high computational complexity. Furthermore, because of the multiobjective nature of this problem, it is impossible to obtain straightforward efficiency from "classical" time series indexing methods. Indeed, these techniques gain most of their pruning power by avoiding computation of irrelevant parts of the search space. It is noteworthy to understand the fundamental differences between multivariate time series problems (extensively studied in literature) and our multiobjective problem. First, multivariate search is a mono-objective problem spanning several dimensions. Therefore, it is equivalent to a multiobjective search with a specific set of weights for all objectives. Hence, this class of problems explores a single direction of optimization. This usually allows to circumvent the problem of pruning power raised by the notion of Pareto dominance, which is the second aspect of computational complexity for our problem. Moreover, multivariate problems usually imply that the series are somehow statistically linked. Oppositely, multiobjective problems allow the optimization of objectives that can be completely decorrelated from each other. Finding the most similar element $\mathcal{S}^*$ to a query corresponds to jointly minimizing the distances between two sets of time series.

$$\mathcal{S}^* = \underset{\mathcal{S}}{argmin} \left\{ \left(D_Q^k \left(\mathcal{S} \right) \right), k = 1, ..., K \right\} \tag{6}$$

As the *ideal point* $\mathcal{S}^*$ which simultaneously optimizes all criteria does not exist, solving this problem turns out to find the set of tradeoff solutions that offer different compromises among objectives. A solution $\mathcal{S}$ is optimal if there is no other solution in the search space that achieves better values than $\mathcal{S}$ on *every* criterion $\mathcal{D}_Q^k \left(\mathcal{S} \right)$. This implies that if we want to know which elements belong to the Pareto front, we should evaluate the distances for all the database and every objective, leading to a brute force analysis. Figure 2 illustrates these concepts. The query is a set of time series input to the system. The first objective is the energy envelope of sounds, and the second is their spectral centroid. The query is at the origin of the criteria space as distances with itself are null in every objective. There is no element in the database that perfectly matches those two properties. Solution $\mathcal{A}$ is the best match for objective $\mathcal{O}_1$. As we can see, its first time series is closely similar to that of the query. Solution $\mathcal{B}$ is respectively the best match for objective $\mathcal{O}_2$. Finally, element $\mathcal{C}$ is the best solution for the associated mono-objective problem with equal weights. We can see that it is not closely similar to any objective, which exhibits the relevance of our approach. Indeed, it allows joint queries on several dimensions without favoring any of them during the search. Furthermore, the multiobjective approach is an appropriate paradigm when the relative weights of each objective are unknown, which is particularly relevant for audio perception [50]. Depending on the problem, regions of the Pareto front might be preferred to others, according to personal preferences.

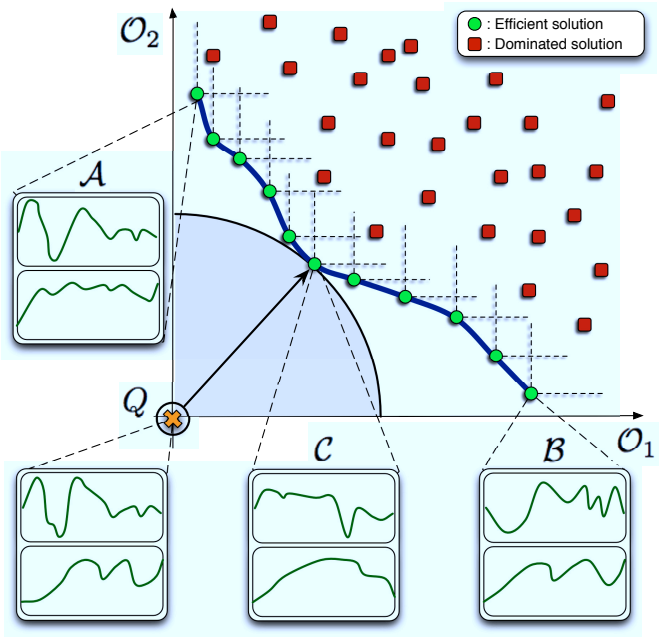

Figure 2. The MOTS matching problem in a bi-objective space. The query Q is at the origin of the space and is represented by a set of time series that have to be matched jointly. We see the results of the system that finds the Pareto solutions. Solution $\mathcal{A}$ is the best match for objective $\mathcal{O}_1$, as we can see the first time series is closely similar to that of the query. Solution $\mathcal{B}$ is respectively the best match for objective $\mathcal{O}_2$. The element $\mathcal{C}$ would be the best monobjective solution. We can see that it is not closely similar to any objective, which motivates the use of multiobjective optimization.

Category	Features
Energy	*EnergyEnvelope, HarmonicEnergy, Loudness, NoiseEnergy, TotalEnergy*
Spectral	*FundamentalFrequency, Inharmonicity, Noisiness, Sharpness, Spread, Flatness, Crest, Centroid, Skewness, Kurtosis, Slope, Decrease, RollOff, Variation*
Harmonic	*Deviation, OddToEvenRatio, Tristimulus, Centroid, Spread, Skewness, Kurtosis, Slope, Decrease, RollOff, Variation*
Sub-bands	*MFCC, RelativeSpecificLoudness, AutoCorrelation, Chroma, ZeroCrossingRate*

Table I
LIST OF AVAILABLE DESCRIPTORS WHOSE MEAN, DEVIATION, TEMPORAL SHAPE AND FIRST AND SECOND DERIVATIVE ARE STORED SEPARATELY. MORE DETAILED INFORMATION CAN BE FOUND IN [56]

B. Audio database description

As we perform queries over large collections of sound samples, we have to maintain a structured database. Figure 3 depicts how sounds are analyzed and managed. We process sound samples with IRCAMDescriptor [56] in order to extract all relevant information from low-level signal data. The list of descriptors used is given in Table I.

The mean and standard deviation of each descriptor are stored in the database. We then normalize the temporal shapes in order to obtain *zero-mean* and *unit-variance*

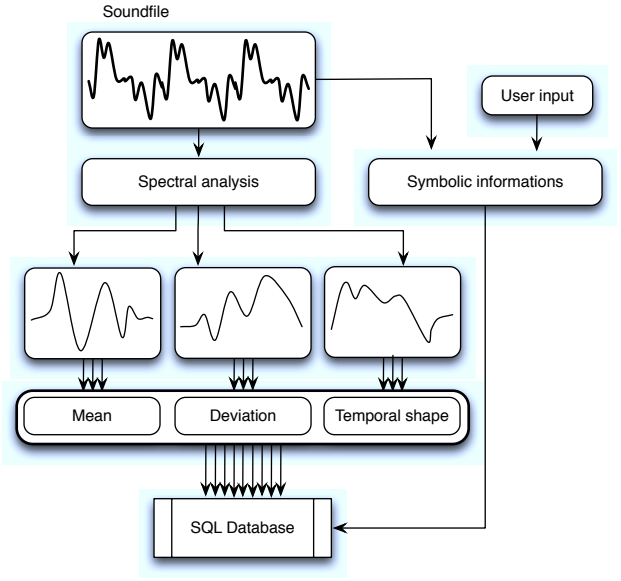

Figure 3. When a soundfile is input to the system, the analysis module computes a set of descriptors whose mean, deviation and temporal shape are stored separately inside an SQL database. Symbolic information can also be stored in the database, either by automatic extraction or direct user input.

time series. We then use the SAX representation [46] to store an efficient modeling of these shapes. Therefore, each element in the database contains several time series which represent different characteristics of a sound. The temporal shapes are resampled to a uniform length. This could be considered as a concern for audio querying as long sounds can be compared to extremely short sounds. However, this approach shows the advantage to focus solely on the temporal shape and the system allows using duration in conjunction with other objectives to be optimized. The duration can alternately be defined as a filtering constraint which will reduce the search space to sounds of matching length. Other symbolic information can also be added to the database. However, we consider in the final search problem that no metadata is available whatsoever. The generic MOTS approach allows joint queries on several temporal properties without favoring any of them. Equipped with an adequate similarity measure along each dimension, we are able to predict various degrees of similarity between the database elements.

C. Algorithms

Because of the ever-growing size of storage capacities, linear scan of an entire database has become unacceptable. Hence, it would be highly desirable to obtain a search method with sublinear time complexity. We introduce two algorithms that can handle the MOTS matching problem. However, because of the novelty of this approach, no competing method exists to evaluate the efficiency of our algorithms. Therefore, the *multiobjective brute force* algorithm will be our testing baseline. This approach requires to compute every distance in each objective, and then extract the Pareto front from the full distance matrix. We describe this reference procedure in Algorithm 1.

Algorithm 1 Brute force multiobjective time series matching algorithm

```
multiobjectiveBruteForce(Q, db)
  for i ∈ [1 ... size(db)]
    for k ∈ [1 ... N_obj]
      compute D_Q^k(S_i)
    end
  end
  P ← extractParetoFront(D_Q(S_j))
end
```

1) Multiobjective early abandon: The complexity of the MOTS problem lies in the repeated computations of time series distances. A natural idea would be to find a way to restrict the amount of distance computations. Instead of computing every distance, we would like to drop calculations as soon as we are confident that the corresponding element is dominated. This technique is known as *early abandon*. However, we have to make fundamental modifications in order to account for the multiobjective nature of our problem. Indeed, early abandon is based on comparing the current similarity against the best distance known so far. However, in a multidimensional context, we cannot simply compare the current distances to a single reference. A turnaround could be to maintain a temporary Pareto front to compare successive elements. However, this approach requires numerous verifications of Pareto dominance, which is an expensive operation. Therefore, it would be preferable to obtain an *approximate distance* for every element beforehand. That way, we could compute only the complete distances of potentially efficient solutions. This approximate distance should be *lower-bounding*, ie. it should underestimate the true distance.

$$\mathcal{D}^k_{approx}(\mathcal{S}_i) \leq \mathcal{D}^k_{true}(\mathcal{S}_i) \ \forall k \in [1, \dots N_{obj}] \qquad (7)$$

With this property, we know that if a set of approximate distances is dominated, then the corresponding set of true distances is dominated. Therefore, we need *simplified representations* of the time series that can provide this approximate distance computation. If these representations are coarse enough, they can account for several time series at the same time. In order to obtain such properties, we can use the SAX representation [48] that perform a temporal and amplitude quantification of the series. In this model, the series are divided into a set of equal-sized temporal steps. Then, the average of the time points contained in each step i is computed and matched to a reduced alphabet.

$$\bar{\mathcal{T}}_i = \alpha\left(\frac{w}{n}\sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i}\mathcal{T}_j\right) \qquad (8)$$

with n the length of the original series, w the number of resulting temporal steps ($w \ll n$) and $\alpha(x)$ a function that matches $x \in \mathbb{R}$ to a discrete alphabet (amplitude quantification). Based on this representation, the iSAX index [67] provides an efficient tree index for time series.

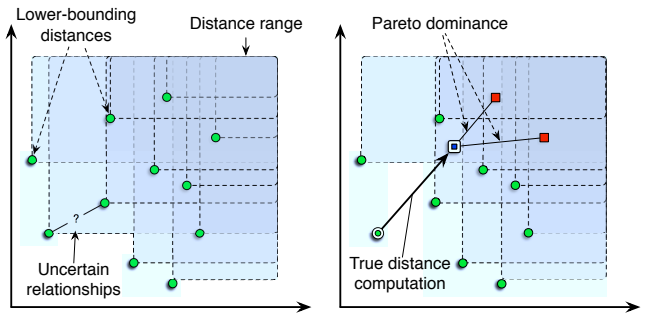

Figure 5. The approximate lower bounding distances in the criteria space and a set of relationships that can or can not be computed

Each level provides a finer representation of the series by increasing the size of the alphabet. Figure 4 illustrates this construction. The series is divided into 8 equal-sized temporal steps. At the first level, the series is quantified by using an alphabet of two elements $\{0, 1\}$. Then, at the subsequent levels, the series are refined by using a larger alphabet $\{00, 01, 10, 11\}$. Obviously, each node in the tree accounts for a whole set of time series. Hence, if we take the first-level of this representation, we obtain a set of *prototypical bins* of reduced cardinality.

Then, the lower-bounding distance between a query Q and a bin representation $\bar{\mathcal{B}}_x$ can be obtained by first transforming the query into the same representation $\bar{Q}$ and then computing

$$\mathcal{D}_{approx}\left(\bar{Q}, \bar{\mathcal{B}}_x\right) = \sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}\left(\mathcal{D}\left(q_i, b_i\right)\right)^2} \qquad (9)$$

Hence, we can obtain the lower bounding positions of every element in the database, as illustrated in Figure 5. It would seem attractive to use these approximate distances to perform a direct assessment of Pareto efficiency. However, these distances are just lower-bound *approximations.* Therefore, the true dominance relations are still uncertain. This is exhibited in Figure 5 with an outlined relationship. It turns out that the distances of a potentially dominating element can be much higher. However, when its true distances are computed, we are sure of its dominance over other elements.

The final implementation is presented in Algorithm 2. We start by transforming the query into the quantified representation. Then, we compute the first level distances for all bins. We store these distances for corresponding elements in the matrix $aDist$. Then, we create an empty Pareto front $\mathcal{P}$ and iterate over the elements of the database. When evaluating an element, as soon as it is dominated by the current front, we abandon computations. If all the distances have been computed, then the current element is potentially efficient. Therefore, we add this element to the current Pareto front and update it accordingly (as the new item might dominate existing solutions in the front). At the end of the algorithm, $\mathcal{P}$ contains the final Pareto front.

Algorithm 2 MOTS matching algorithm with early abandon

```
multiobjectiveEarlyAbandon(Q, db, idx)
  // Quantify the query
  Q̄^{k∈[1...N_obj]} = { (w/n) Σ_{j=(n/w)(i-1)+1}^{(n/w)i} Q_j^k, i ∈ [1...w] }
  // Compute query-to-bins distances
  for b ∈ [1...N_bins^k]
    aDist^k_{i∈B_b^k} = D_approx(Q̄^k, B̄_b^k)
  end
  P = ∅
  // Perform multiobjective abandon
  for i ∈ [1...size(db)]
    for k ∈ [1...N_obj]
      if isDominated(aDist_i, P)
        abandon;
      else
        aDist_i^k = D_Q^k(S_i)
    end
    add(S_i, P);
    P = extractParetoFront(P);
  end
```

2) Hyperplane search: The main drawback of the previous algorithm is that it still requires frequent Pareto optimality assessments. Hence, it would be wiser to find a less expensive theoretical limit to drop computations of the distance measures. Therefore, our main idea is to construct an approximate Pareto hyperplane $\mathcal{P}$ to act as our theoretical limit. We can obtain this hyperplane by using 1-NN queries from efficient time series indexing for each objective. These queries will give us boundary elements of the final Pareto front. This is straightforward from the fact that these elements cannot be dominated as they have the smallest distance in one of the objectives.

$$\forall \mathcal{S}_i, \exists k \mid \forall \mathcal{S}_j, \mathcal{D}_Q^k\left(\mathcal{S}_i\right) < \mathcal{D}_Q^k\left(\mathcal{S}_j\right) \Rightarrow \mathcal{S}_i \in \mathcal{P} \qquad (10)$$

Hence, we can prune elements whose approximate distances are dominated by this hyperplane. This can be computed straightforwardly if we obtain the hyperplane normal. We show how to compute this normal efficiently by avoiding an expensive least-squares minimization.

Proposition 3. *Given a nonzero vector* $\mathbf{n}$ *in* $\mathbb{R}^m$ *and a point* $\mathbf{p} \in \mathbb{R}^m$*, the hyperplane perpendicular to* $\mathbf{n}$ *through* $\mathbf{p}$ *is the set of all* $\mathbf{x} \in \mathbb{R}^m$ *such that* $(\mathbf{x} - \mathbf{p}) \cdot \mathbf{n} = 0$.

Therefore, if we want to find the normal of hyperplane $\mathcal{H}$, we must find the vector $\mathbf{n}_p \in \mathbb{R}^m$ satisfying $\mathcal{P}\mathbf{n}_p = \mathbf{0_m}$ where $\mathcal{P}$ is a $k \times m$ matrix and $\mathbf{0_m}$ is a $m \times 1$ zero vector. $\mathcal{P} = [p_1, \ldots, p_k]$ is the set of Pareto points defining the hyperplane $\mathcal{H}$ (in our case p_i will be the 1-NN result for the i^{th} objective). In order to obtain this vector, we must solve

$$\mathbf{n}_p = \underset{\mathbf{v}}{argmin}\left(\mathbf{v}^T\mathcal{P}^T\mathcal{P}\mathbf{v}\right) \qquad (11)$$

Alone, this equation yields the trivial solution $\mathbf{n}_p = \mathbf{0_m}$

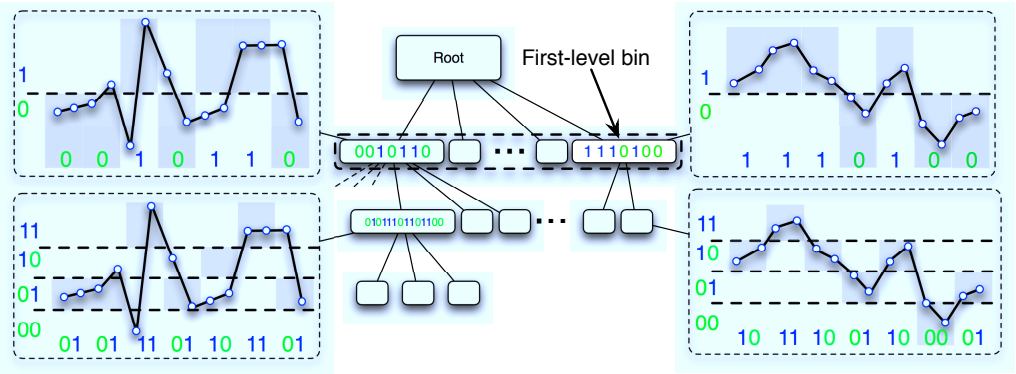

Figure 4. Construction of the quantified bins for time series and computation of the 1st-level distance for a query time series.

which we obviously want to avoid. To avoid this case, we can add the constraint $\|\mathbf{n}_p\| = 1$, which can be rewritten as $1 - \mathbf{n}_p^T \mathbf{n}_p = 0$. Therefore, in order to find the best value for $\mathbf{n}_p$, we can use the Lagrange multipliers and solve

$$\frac{\delta}{\delta \mathbf{n}_p} \left(\mathbf{n}_p^T \mathcal{P}^T \mathcal{P} \mathbf{n}_p + \lambda \left(1 - \mathbf{n}_p^T \mathbf{n}_p \right) \right) = 0 \qquad (12)$$

After applying the derivation, we obtain the characteristic equation $\left(\mathcal{P}^T \mathcal{P} - \lambda E \right) \mathbf{n}_p = 0$. Therefore, we know that $\mathbf{n}_p$ is an eigenvector of $\left(\mathcal{P}^T \mathcal{P} \right)$ and λ is an eigenvalue. However, we can not control the orientation of the normal (as any hyperplane possess two oppositely oriented normal vectors). Furthermore, this also requires to compute some eigenvectors with potentially large dimensionality which can be expensive. In order to alleviate both problems at the same time, we have to modify the original constraint slightly. For that purpose, we introduce a direction vector $\mathbf{d}$ that will ensure the orientation of the normal vector. Therefore, we constrain the normal vector $\mathbf{n}_p$ to have the same orientation as $\mathbf{d}$. We can write this constraint as $\left(1 - \mathbf{d}^T \mathbf{n}_p \right)^2 = 0$. Hence, we must now solve

$$\mathbf{n}_p = \underset{\mathbf{v}}{argmin} \left(\mathbf{v}^T \mathcal{P}^T \mathcal{P} \mathbf{v} + \left(1 - \mathbf{d}^T \mathbf{v} \right)^2 \right) \qquad (13)$$

By using the same reasoning than previously, we can find the extreme value by solving

$$\frac{\delta}{\delta \mathbf{n}_p} \left(\mathbf{n}_p^T \mathcal{P}^T \mathcal{P} \mathbf{n}_p + \left(1 - \mathbf{d}^T \mathbf{n}_p \right)^2 \right) = 0 \qquad (14)$$

Therefore, by taking the same matrix derivatives and simplifying, we obtain the normal by computing

$$\mathbf{n}_p = \left([\mathcal{P}, \mathbf{d}] \, [\mathcal{P}, \mathbf{d}]^T \right)^{-1} \mathbf{d} \qquad (15)$$

where $[\mathcal{P}, \mathbf{d}]$ is the matrix obtained by concatenating matrix $\mathcal{P}$. In our implementation, we use $\mathbf{d} = \underset{j}{max}(p_i^j)$, $p_i \in \mathcal{P}$ to ensure the orientation of the resulting normal.

Proposition 4. *Let $\mathcal{P}$ be the hyperplane of all $\mathbf{x} \in \mathbb{R}^k$ with $(\mathbf{x} - \mathbf{p}) \cdot \mathbf{n} = 0$ such that $\mathbf{n} \neq \mathbf{0}$. Then the distance of any point $\mathbf{a}_x \in \mathbb{R}^k$ from the hyperplane $\mathcal{P}$ is given by*

$$dist\left(\mathbf{a}_x, \mathcal{P}\right) = \frac{(\mathbf{a}_x - \mathbf{p}_i) \cdot \mathbf{n}}{\|\mathbf{n}\|} \qquad (16)$$

with $\|\mathbf{n}\|$ the norm of the normal $\mathbf{n}$ and $\mathbf{p}_i \in \mathcal{P}$ is one of the Pareto points.

The final implementation is presented in Algorithm 3 and illustrated geometrically in Figure 6. The implementation presented here can be used with any representation, distance and indexing techniques available. We simply assume that a time series index is constructed for each objective in order to perform efficient 1-NN queries and, therefore, avoid linear scan. We also consider that the index provides a lower bounding distance measure on indexing nodes. Given a query Q, a database `db` and a set of index `TS-Indexes` for each objective (constructed prior to the search), we start by transforming the query and computing the first-level distances as previously. This set $aDist$ is then used to perform the 1-NN exact queries on each objective. These queries give us the initial Pareto front $\mathcal{P}$ that form the approximate Pareto hyperplane. The 1-NN queries also compute a small portion of exact distances for each objective that we recover in list `aDist`. That way, after 1-NN queries we already have an approximate lower bounding position for each element. We then obtain the normal of the hyperplane defined by the list of Pareto points. Then, we evaluate each element of the database and stop distance computation as soon as they are dominated by the hyperplane. If we compute the complete distances in every objective for an element, we add it to the list of potential Pareto points. Finally, we filter this list by extracting the final Pareto front $\mathcal{P}$.

D. Efficiency on massive databases

We evaluate the efficiency of our algorithms against the *multiobjective brute force* algorithm on synthetic and real datasets. The artificial dataset is composed of random walk time series generated with a constant size of 512 time points. An independent set is synthesized for each hypothetical objective. The real dataset is a combination

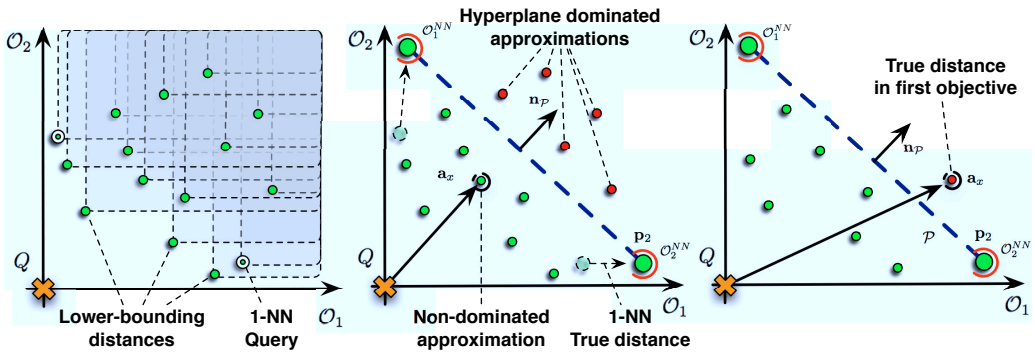

Figure 6. Geometric interpretation of the *multiobjective hyperplane search* algorithm

Algorithm 3 MOTS matching algorithm by approximate hyperplane search.

multiobjectiveHyperplaneSearch$(Q,$ db$)$
  // Quantify the query
  $\bar{\mathcal{Q}}^{k\in[1\ldots N_{obj}]} = \left\{ \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} \mathcal{Q}_j^k, i \in [1 \ldots w] \right\}$
  // Compute query-to-bin distances
  for $k \in [1 \ldots N_{obj}]$
    for $b \in \left[1 \ldots N_{bins}^k\right]$
      $aDist_{i\in\mathcal{B}_b^k}^k = \mathcal{D}_{approx}\left(\bar{\mathcal{Q}}^k, \bar{\mathcal{B}}_b^k\right)$
    end
  end
  // Perform efficient 1-NN queries
  $[\mathcal{P}\ aDist]$ = 1NN-Queries$(Q,\ aDist,$ TS-Indexes)
  // Reference direction vector
  $\mathbf{d} = \underset{j}{max}(p_i^j),\ p_i \in \mathcal{P}$
  // Compute hyperplane normal
  $\mathbf{n}_p = \left([\mathcal{P}, \mathbf{d}]\,[\mathcal{P}, \mathbf{d}]^T\right)^{-1} \mathbf{d}$
  // Transform into unit-norm vector
  $\mathbf{n}_p = {}^{\mathbf{n}_p}/\sqrt{\mathbf{n}_p^T \mathbf{n}_p}$
  for $i \in [1 \ldots size\,(db)]$
    for $k \in [1 \ldots N_{obj}]$
      if $(aDist_i - p_1) \cdot \mathbf{n}_p < 0$
        abandon
      else
        $aDist_i^k = \mathcal{D}_Q^k\,(\mathcal{S}_i)$
    end
    add$(\mathcal{S}_i,\ \mathcal{P})$
  end
  checkParetoFront$(\mathcal{P})$

of *Studio On Line* [4], *Real World Computing* [28] and *Vienna Symphonic Library* instrumental databases. These datasets include single notes of different playing modes from 23 orchestral instruments, which amounts to a total of 213.814 sound files. These files are WAVE and AIFF format, quantified to 16-bit at a sampling rate of 44.1 kHz. Subsets of the collections are randomly selected for increasing database sizes. Objectives are also randomly selected from the set of audio descriptors (cf. Table I). This selection procedure is ten-folded. For each set of parameters, one hundred queries are processed in order to avoid statistical anomalies. Queries are random walk time series with a constant size of 512 points. Computations were performed on a Macbook 2.4 GHz Dual Core running under Mac OS X 10.6.6 with 2 GO of DDR3 RAM.

We present the results of *querying wall time* for synthetic datasets in Figure 7. The left figure shows the *median* (dotted line), *average* and *variance* (solid line) in querying time for increasing database sizes. As we can see, the early abandon algorithm provides up to two times of speedup over the brute force approach, with a low variance in querying time. However, this factor of speedup is linear to the cardinality of the dataset. The *hyperplane* algorithm is strongly superior, as it provides up to ten times of speedup over the brute force approach. The differences between the early abandon and hyperplane approaches can be explained by the higher number of Pareto front evaluations in the first one. However, the variance of the hyperplane search also increase with the cardinality, which imply that the querying time might vary more importantly. Analysis of results reveals that both algorithms performs better on real sound collections. This could be explained by the distribution of time series in real datasets, which is unlikely to be uniform as it is for random walk datasets. The most enthralling finding concerns the efficiency on an increasing number of objectives, presented in Figure 7 (right). As we can see, the early abandon still provides a linear factor of speedup. However, the hyperplane algorithm exhibits a sublinear behavior with a significantly lower median. This behavior could be explained by the higher probability that a large portion of the space is ruled out by the approximate hyperplane in higher dimensions.

To analyze this hypothesis, we compare the pruning power induced by each algorithm. The *space pruning ratio* is computed by comparing the proportion of distances that are not evaluated to the quantity of points in the dataset. The main advantage of this measure is that it is hardware independent and is also independent of the distance measure used. Figure 8 (left) exhibits the space pruning ratio provided by the algorithms for an increasing database cardinality. As we can see, the *hyperplane* limit provides a strongly superior pruning ratio. The variances seem to remain almost constant for both algorithms, with a higher variance for the hyperplane algorithm. However,

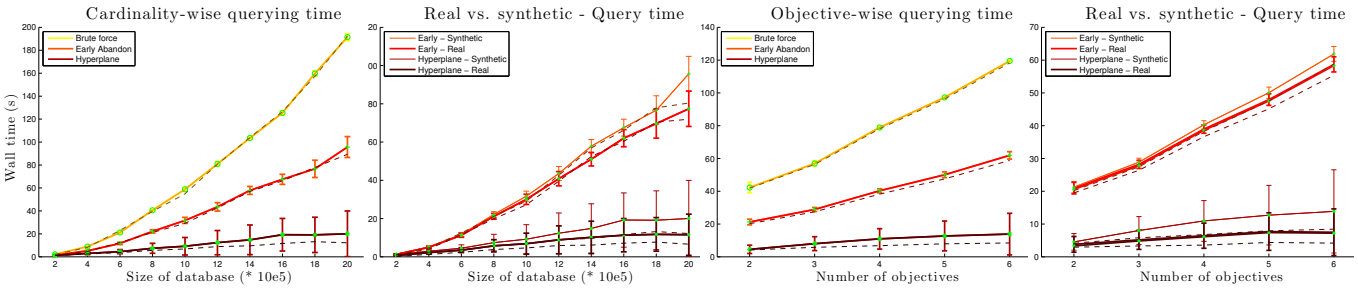

Figure 7. Query wall time (in seconds) for increasing database size (left) and increasing number of objectives (right) on synthetic and real datasets.

an equivalent variance for higher cardinality will imply a higher variance in the number of elements pruned. In both case, the techniques seem to indicate an upper bound in pruning power as the number of time series increases. Figure 8 (right) exhibits the space pruning ratio provided for a growing number of objectives. As we can see, the hyperplane algorithm quickly converge to a constant pruning ratio (which can explain its sublinear time complexity), whereas the early abandon algorithm exhibits a continuous drop in pruning power.

IV. Classification tasks

A. Classification selection criterion

The MOTS approach allows to find the set of efficient solutions given an audio query and multiple optimization features. In the multiobjective framework, there is no way to order the Pareto front. Therefore, possibilities cannot be ranked among each other. However, to assess the quality of the proposed approach, we evaluate it in classification tasks. Therefore, we need a criterion to make the final classification decision, ie. to select which class is the best match to a given input. We introduce in this section two new class selection criteria.

1) Pareto cardinality: Given the Pareto set, we can first simply look at its cardinality. Therefore, our first selection criterion is obtained by counting the number of occurrences of each class in the Pareto front. The selected class is the most represented in the front. This is obviously a basic criterion and we can expect it to be less efficient in higher dimensions. We term this method *MOTS* in the following.

2) Hypervolume domination: We introduce a novel criterion based on *hypervolume* domination. This measure has been used in multiobjective optimization with Genetic Algorithms (GA) [87] as a performance indicator, i.e. only to differentiate the quality of different algorithm. However, it has never been used as a classification criterion to our best knowledge. The idea behind this measure is that every point in a multidimensional space, defines a hypervolume which indicates the portion of space dominated by this point. For a n-dimensional space, $n \in \mathbb{N}$, the hypervolume of a box in $\mathbb{R}^n$ generated by two points $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ is defined as

$$\mathcal{H}(B) = \prod_{i=1}^{n} (b_i - a_i) \tag{17}$$

The *hypervolume dominated* by a Pareto front $\mathcal{P}$ given a reference point $r_p = \left(r_p^1, \ldots, r_p^n\right)$ is given by the union of the hypervolumes dominated by each point in the front

$$\mathcal{H}(\mathcal{P}) = \mathcal{H}\left(\bigcup_i B_i\right) = \mathcal{H}\left(\bigcup_{(p_1,\ldots,p_k)\in\mathcal{P}} [p_1, r_p] \times \ldots \times [p_k, r_p]\right) \tag{18}$$

These notions are shown in Figure 9 (up). Point p_1 defines a box $\mathcal{B}_1$ (darker gray) with the reference point r_p. Each point of this set also implies a corresponding domination box. Therefore, the hypervolume dominated by the Pareto front is the union of hypervolumes dominated by each point in the front.

The interest of working with the hypervolume is that it provides a total order among sets of points that are normally only partially comparable through the Pareto dominance. Hence, the hypervolume indicator has several interesting properties regarding the total ordering it provides. First, it is a refinement of the Pareto dominance relation. Hence, maximizing this indicator results in Pareto-efficient solutions only [88]. It has also be shown to be the only unary indicator to detect the weak dominance relation between any two sets of solutions [87]. As for classification decisions, we need a unary measure this further motivates the interest of using the hypervolume indicator, as it is the only Pareto-compliant indicator. This means that the hypervolume is the only measure to provide a ranking of sets which does not contradict the Pareto dominance relations. Other interesting results shows that the hypervolume indicator guarantees strict monotonicity and provides the best possible approximation ratio for linear and concave fronts [26]. Figure 9 shows the benefits of such a measure when comparing two distributions. Even if the first class has more elements belong to the final Pareto front, its dominated hypervolume $\mathcal{H}_1$ is smaller than the hypervolume $\mathcal{H}_2$ of the second class. Therefore, the hypervolume indicates both the fitness of a distribution and its spread over the optimization space. Furthermore, compared to a NN or NC rule, it summarizes the behavior of the whole class with respect to the input rather than the position of the input relative to the elements of the class. This means that the hypervolume provides a measure of the three criterion for the quality of multiobjective sets : (i) closeness of solutions to the origin, (ii) good

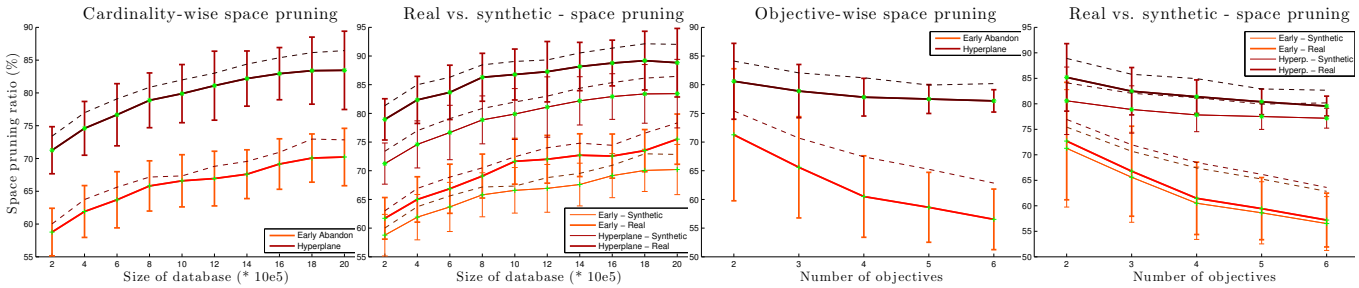

Figure 8. Space pruning ratio for increasing database size (left) and number of objectives (real) on synthetic and real datasets.

distribution and spread along the front and (iii) number of non-dominated points [7]. In our implementation, we use the hypervolume computation algorithm proposed by [25]. We compute the hypervolume dominated by the Pareto front of each class. Hence, the selected class is the one which induces the largest dominated hypervolume. We term this approach *HyperVolume-MOTS* (HV-MOTS).

B. Classification results

In order to assess the performance of our approach, we evaluate it in classification tasks using two datasets. First, the reference MuscleFish dataset [79] allows to compare our approach to state-of-art methods. Second, we collected a more recent and comprehensive dataset to test how our approach scales up to wider sets of data. It should be noted that all the sounds from the datasets used in this paper are isolated single-source clips. Both datasets are available on a supporting web page dedicated to this paper [1] so that the results of our experiments are fully reproducible.

1) MuscleFish: This dataset, assembled by Wold et. al [79], has been used extensively [32], [31], [44], [63], [65] in order to compare performances of different systems. It is composed of 409 sound files which are divided into 16 classes. Complete description of the dataset is presented in Table II. Files are single-channel Sun/Next (.au) μ-law encoded audio files quantized to 8-bit with a sampling rate of 8 kHz. Loudness levels and file lengths vary over samples with the average size of a file being about 50 KBytes.

2) Freesound: In order to evaluate how our approach scales up to more comprehensive datasets, we collected 2193 sounds representing 54 classes from the Freesound project [2], which makes this set five times larger than the MuscleFish dataset. Complete description is presented in Table III. Files are single and double channels, WAVE and AIFF format, quantized to a minimum resolution of 16-bit with a minimum sampling rate of 44.1 kHz. Loudness levels and file lengths vary with the average size of a file being about 310 KBytes.

3) Evaluation methodology: The goal of the classification task is to input a sound file into the system which tries to find which class it belongs to. As our method does not require any training, we use the *Leave-One-Out* evaluation methodology. That is, each file is first withdrawn from the dataset and then input for classification with the remaining set acting as a database. In order to measure performances, we use the *classification accuracy* defined as $\mathcal{A}_{cl} = {}^{N_{true}}/_{N}$ with N_{true} the number of clips correctly classified and N the total number of clips in the dataset. In order to compare different methods, we performed large-scale experiments by testing combinatorial possibilities among available descriptors. Therefore, we start by testing classification accuracy for every single descriptor listed in Table I. Then, we evaluate the classification with every combination of two descriptors, and so forth. Given that this testing methodology implies an exponentially growing number of tests, we keep only the top performing half of the descriptors after each step, based on their classification accuracies. We repeat this procedure and halve the set of available descriptors (in which to choose the objectives for classification) until the number of remaining descriptors is less than the number of objectives. As an exhaustive analysis of the discriminative power for each feature is beyond the scope of this article, complete results of classification tasks are available on the supporting web page dedicated to this paper.

We are comparing the HV-MOTS classifier against the 1-NN and 5-NN mono-objective selections. This comparison is based on the same set of time series and mean features for all classification criterion. Even if these methods might seem easy to overpower, several published studies confirm

Musical instruments		**Effects**	
Altotrombone	13	Animals	9
Bells	7	Crowds	4
Cellobowed	47	Laughter	7
Oboe	32	Machines	11
Tublarbells	19	Percussion	99
Violinbowed	45	Telephone	17
Violinpizz	40	Water	7
Speech			
Female	35	Male	17
		Total	409

Table II
DESCRIPTION OF THE MUSCLEFISH DATASET [79] USED IN CLASSIFICATION TASKS. 409 SOUNDS ARE DIVIDED INTO 16 CLASSES.

[1] http://repmus.ircam.fr/esling/ieee-mots.html
[2] http://www.freesound.org

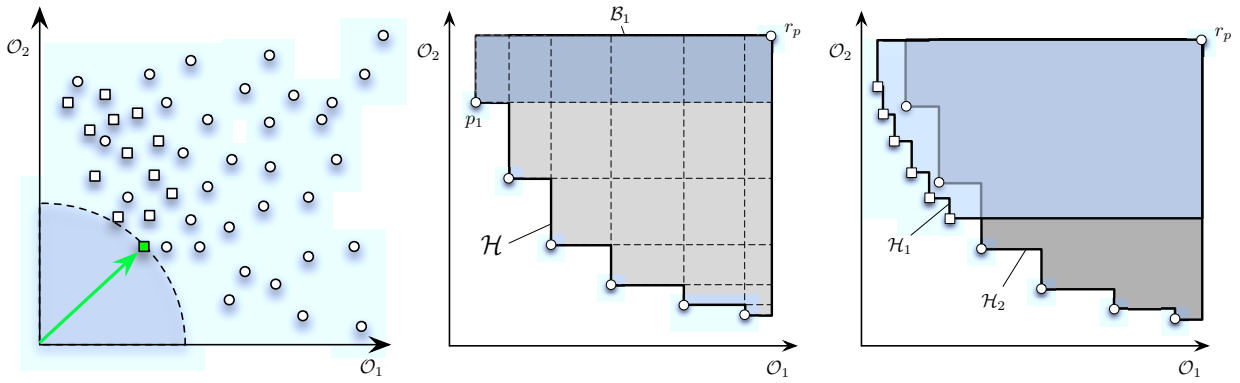

Figure 9. (Left) The nearest neighbor to the origin (query) in distance space based on the Euclidean distance in two dimensions. (Middle) Hypervolume dominated by a Pareto front given the reference point r_p in a 2-dimensional space. The darker gray subpart defines the box $\mathcal{B}_1$ which is dominated by point p_1. The hypervolume $\mathcal{H}$ dominated by the Pareto front is defined as the union of all boxes dominated by each point of the front. (Right) Comparison of two dominated hypervolumes $\mathcal{H}_1$ and $\mathcal{H}_2$. Even though the first class have more elements belong to the final Pareto set, its hypervolume $\mathcal{H}_1$ is smaller than the hypervolume $\mathcal{H}_2$ of the second class.

Western instruments		**Indian instruments**		**Animals**		**Effects**	
Alto-flute	85	Santoor	15	Birds	29	Applause	41
Bassoon	80	Singing-Bowl	8	Cat	26	Footsteps	33
Cello	59	Tabla	18	Dog	21	Gunshots	15
Clarinet	76	Tambura	16	Horse	18	Heartbeats	23
Contrabass	66	Thumb-piano	17	**Synthesis**		Laughter	141
Glockenspiel	17	**Drums**		Bassline	99	Musicbox	28
Guitar (dist)	16	Crash	43	Reese	47	Paper	17
Oboe	113	Hi-hats	25	Vocoder	43	Siren	37
Saxophone	27	Kick	27	Wobble	39	Subway	11
Trombone	42	Loops	25	**Speech**		Sword	21
Trumpet	35	Snare	42	Female	96	Telephone	16
Tuba	74	Scratch	20	Male	87	Thunder	29
Viola	58	Toms	9	Robotic	31	Water	43
Violin	98	Tone	8	Scream	32	Whistle	26
						Zipper	17
						Total	2193

Table III
DESCRIPTION OF THE FREESOUND DATASET COLLECTED SPECIFICALLY FOR OUR STUDY. 2193 SOUNDS ARE DIVIDED INTO 54 CLASSES.

that 1-NN selection is still by far the top performing classification scheme for time series data [30], [36], [60]. Some authors point out "*while there have been attempts to classify time series with decision trees, neural networks, Bayesian networks, support vector machines etc., the best published results (by a large margin) come from simple 1-NN methods*" [20]. Even the SVM classifier has been shown to be *at most* statistically equivalent to 1-NN but usually performs worse [30]. This explains why we focus our comparison on the 1-NN classifier.

C. Results analysis

We present in Figure 10 the classification accuracies on the MuscleFish dataset for a growing number of objectives. For a given number of objectives, the top figure provides the *mean* accuracy over every combination and the figure below is the *best* score obtained by a single combination. As we can see, the HV-MOTS classifier consistently outperforms the other approaches in classification accuracy. This result is confirmed by the accuracies obtained on the Freesound dataset, presented in Figure 11. Even with up to five times more classes and sounds, HV-MOTS exhibits an almost equivalent classification accuracy and still outperforms other methods.

More interestingly, it seems that HV-MOTS strongly outperforms other approaches in *mean* classification accuracy. This implies that given any set of features, the multiobjective approach will obtain better results. To support this claim, we provide in Figure 12 the results of statistical significance tests between methods and across datasets, to rule out the effect of a particular data distribution. We use Tukey-Kramer Honestly Significant Difference (HSD) test [17] over the results of Friedman's ANOVA to see if one method is statistically significantly different from the rest. We also present the *statistical mean accuracy* computed with a one-way ANOVA. Finally, we present

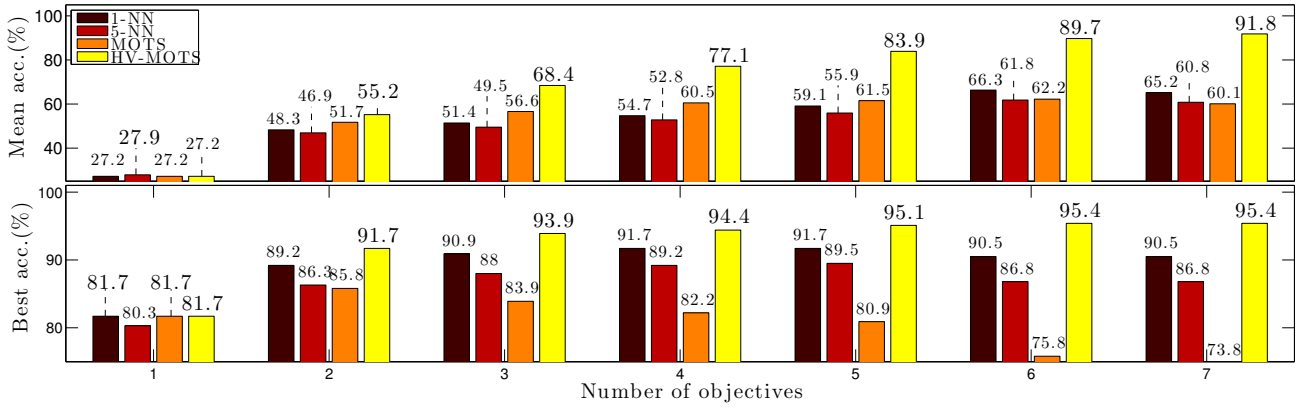

Figure 10. Classification results on the MuscleFish dataset for a growing number of objectives. For a given number of objectives, the left column indicates the mean classification accuracy and the right column indicates the best classification accuracy.

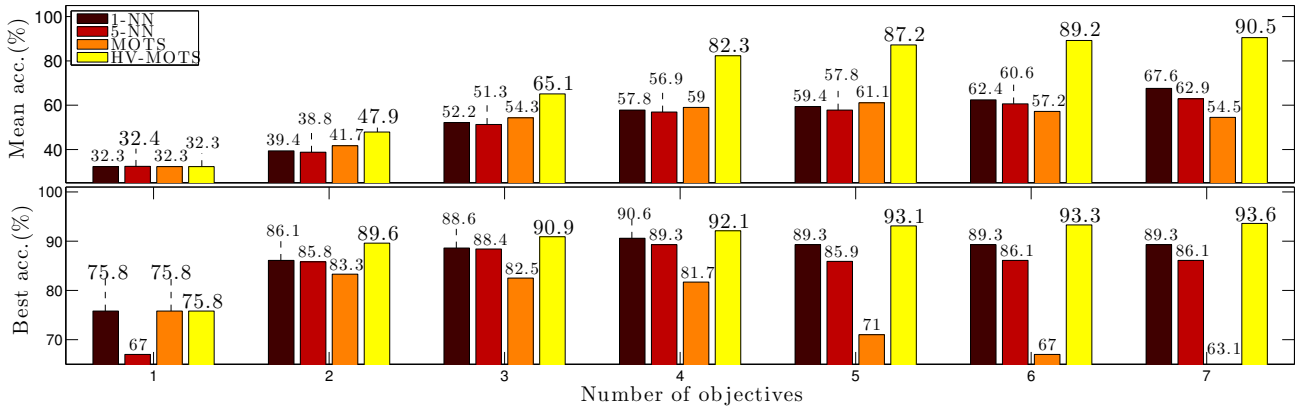

Figure 11. Classification results on the Freesound dataset for a growing number of objectives.

the *critical difference graphs* [19] which allows to exhibit the true statistical superiority and eventual groups of statistical equivalence between various methods. This graph summarizes the column ranking of all methods over *every* features combinations. We can see in this figure that the mean column ranks and statistical mean difference in accuracy of HV-MOTS are strongly superior. The column rank corresponds here to the ranking of methods based on their accuracy results. This means that, after two objectives, the HV-MOTS method is almost always in first position for any descriptor combination if ranked against other methods based on their accuracy score. Furthermore, the mean differences in accuracy increase with the number of objectives. It seems that the multiobjective classification is able to maintain the discriminative power of the best feature involved, whereas mono-objective selection will be confined by the worst features. This may go against the hypothesis that the feature set is more important than a particular learning scheme [52]. Furthermore, it seems here that the behavior of the whole class with respect to the input may be more influential than the position of the input relative to the elements of the class. Therefore, even with lower dimensionality involved, the multiobjective paradigm is able to achieve a satisfactory classification accuracy. We can see that the MOTS paradigm (based on Pareto cardinality) is superior in mean classification accuracy to mono-objective selections for low dimensionality but starts to regress after four dimensions. This may come from the fact that an increasing number of dimensions creates more inclusive Pareto fronts which deludes the cardinality indicator. For other methods, performances stabilize and even regress slightly after five dimensions are involved.

We present in Table IV the confusion matrix of the best classification accuracy (95.4%) obtained by HV-MOTS on the MuscleFish dataset. The corresponding descriptor combination is composed of MFCC, MFCCDeltaStdDev, PerceptualSlope, ChromaDeltaStdDev, RelativeSpecificLoudnessDeltaStdDev and PerceptualDecrease. It is interesting to note that most of the features used are related to the temporal behavior of the sound spectrum. Even the average descriptors are deviations of derivative, which, in fact, summarize the quantity of temporal variations for these descriptors. Furthermore, this combination contains descriptors for each structural aspects of sounds, namely energy (*Loudness*), harmony (*Chroma*), spectral shape *(MFCC)* and perceptual descriptors. It should be noted that the same accuracy was obtained by 18 similar combinations (which further confirms our intuition that HV-

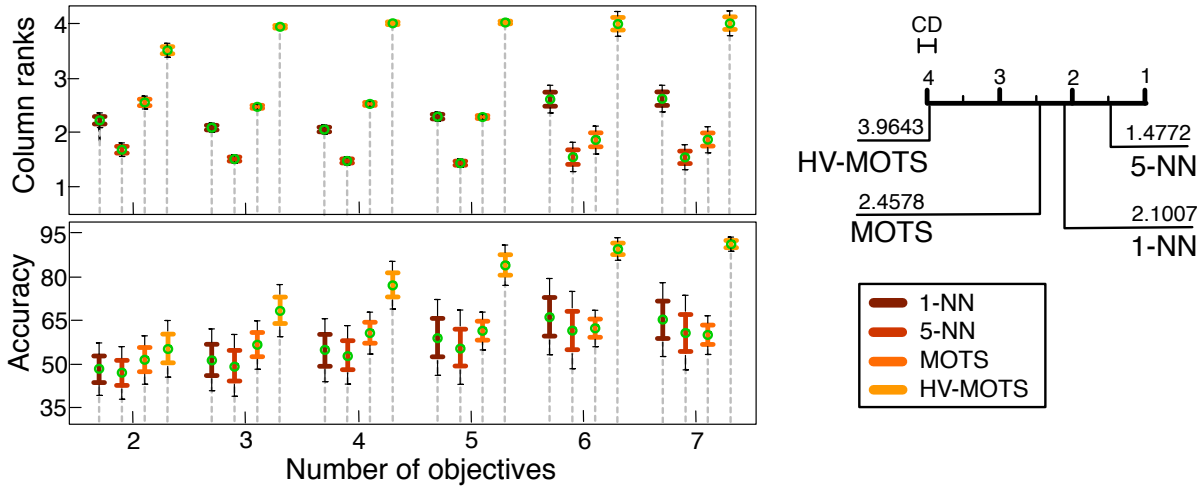

Figure 12. Significance tests between various methods for a growing number of objectives across both datasets. For a given number of objectives, the left column indicates the mean column rank (from Tukey-Kramer HSD over Friedman's ANOVA) and the right column gives the statistical mean difference in accuracy with the top performing method (from a one-way ANOVA).

MOTS is able to retain the discriminative power of the best features involved). If we look at the distribution of the confusion matrix, we can outline different types of errors made by the system. First, the *class similarity* errors that can be expected when similar classes are part of datasets with widely diverse class types. For instance, elements of *male speech* are confused for *female speech* and the same apply to *violinbowed* confused with *cellobowed*. Second, the *morphological similarity* errors can be observed when the spectral behavior of two classes is alike. For instance, some *violinpizz* are confused with *percussions* because of the impulsive nature of such sounds. The same applies to *machines* confused with *water* because of the long-term repetitive patterns that emerge from both. Finally, in both types can be found some *reciprocal errors* where the error applies symmetrically to two classes.

The HV-MOTS method was designed based on the hypotheses that temporal shapes would improve static information and at the same time multiobjective selection would provide a perceptually more relevant and, therefore, more accurate classification. In order to analyze these hypotheses, we confront different views on experimental results. Figure 13 provides a comparison of the classification accuracy of using *only* temporal features, *only* static (mean and deviation) features or mixed sets of information. As we can see, the use of temporal features performs better than static features. More interestingly, it appears that best results are obtained by mixed sets of information, which indicates that normalized temporal shapes and static information are complementary sets of information. Finally, for any type of descriptors used, multiobjective selections perform consistently better than mono-objective approaches.

D. Comparison to state of the art

We compare our results to the state-of-art methods proposed with the same evaluation framework, namely a classification task on the MuscleFish dataset with a *Leave-One-Out* methodology. This allows to report published classification accuracies as a baseline for comparison. In their original study, Wold et al. [79] proposed to compute the mean, variance and autocorrelation of loudness, pitch, brightness and bandwidth, which together with duration amounts to a total of 13 features. By using a 1-NN rule, comparing the query to all feature vectors in the database with the Euclidean distance, they reported 80.9% classification accuracy. Guo et al. [32] later tested the applicability of a machine learning technique called *Boosting* based on a vector of 8 perceptual cepstral features which provided 78.3% accuracy. Guo and Li [31] proposed to use SVM on the same feature set and obtained 89% accuracy. Li [44] introduced the NFL method which was shown to provide 90.22% accuracy. Reyes-Gomes and Ellis [63] studied the use of GMM-EM and HMM with low entropy learning and obtained 89.9% accuracy. Finally, Shao et al. [65] used Neural Networks trained by GA with Back Propagation (BP-GA) over a set of 17 features and reported 92% classification accuracy. However, their results are based on separate *Train* and *Test* sets procedure which does not allow straightforward comparison. The HV-MOTS method allows to obtain 95.35% classification accuracy, which outperforms previously reported accuracies for this dataset. Table V synthesizes the comparison between our method and previous approaches.

E. Robustness analysis

In real-life conditions, we can expect audio collections to include sounds from different sources recorded under various conditions. Some QBE systems have been tested for robustness but usually only with regards to transcoding, using either lower sampling rates [33] or lossy data compressions [9] to simulate mobile audio databases. We test our approach by applying a wider range of distortion classes to simulate various low-quality conditions in recording

	Altotrombone	Animals	Bells	Cellobowed	Crowds	Laughter	Machines	Oboe	Percussion	Speech (female)	Speech (male)	Telephone	Tuba	Violinbowed	Violinpizz	Water
Altotrombone	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Animals	0	**8**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0
Bells	0	0	**6**	0	0	0	0	0	0	0	0	0	0	0	**1**	0
Cellobowed	0	0	0	**47**	0	0	0	0	0	0	0	0	0	0	0	0
Crowds	0	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0
Laughter	0	0	0	0	0	**7**	0	0	0	0	0	0	0	0	0	0
Machines	0	0	0	0	0	0	**8**	0	0	0	0	0	0	0	0	**3**
Oboe	0	0	0	0	0	0	0	**32**	0	0	0	0	0	0	0	0
Percussion	0	0	0	0	0	0	0	**1**	**97**	0	0	0	0	0	**1**	0
Speech (female)	0	0	0	0	0	0	0	0	0	**35**	0	0	0	0	0	0
Speech (male)	0	0	0	0	0	0	**1**	0	0	**3**	**12**	0	0	0	0	**1**
Telephone	0	0	0	0	0	0	0	0	**1**	0	0	**16**	0	0	0	0
Tublarbells	0	0	0	0	0	0	0	0	0	0	0	0	**19**	0	0	0
Violinbowed	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**44**	0	0
Violinpizz	0	0	0	0	0	0	0	0	**2**	0	0	0	0	0	**38**	0
Water	0	0	0	0	**1**	0	**1**	0	0	**1**	0	0	0	0	0	**4**

Table IV
CONFUSION MATRIX FOR THE BEST CLASSIFICATION ACCURACY (95.4%) OBTAINED BY HV-MOTS ON THE MUSCLEFISH DATASET. THE DESCRIPTOR COMBINATION USED IS COMPOSED OF MFCC, MFCCDELTASTDDEV, PERCEPTUALSLOPE, CHROMADELTASTDDEV, RELATIVESPECIFICLOUDNESSDELTASTDDEV AND PERCEPTUALDECREASE.

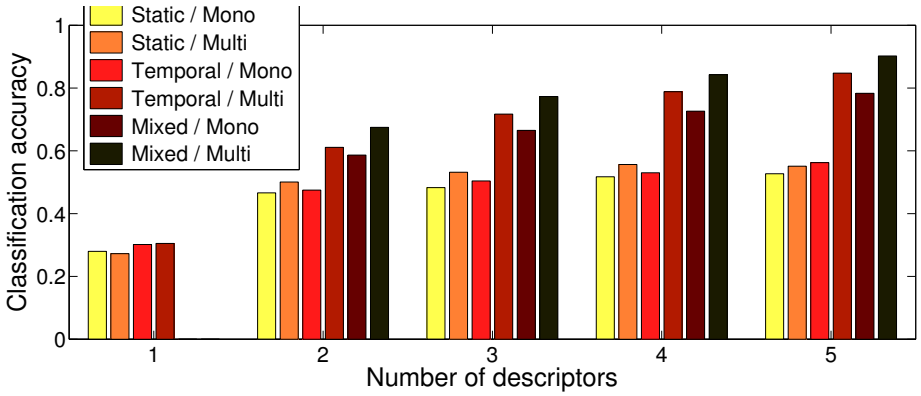

Figure 13. Comparison of the classification accuracy of using *only* temporal features, *only* static (mean and deviation) features or *mixed* sets of information with either multiobjective or mono-objective selection.

	Accuracy	N
Guo et al. [32]	78.3 %	8
Wold et al. [79]	80.9 %	13
Guo and Li [31]	89.0 %	8
Reyes-Gomes [63]	89.9 %	-
Li [44]	90.2 %	8
HV-MOTS	**95.4 %**	**6**

Table V
COMPARISON OF OUR METHOD TO STATE-OF-ART METHODS ON THE MUSCLEFISH DATASET WITH A *Leave-One-Out* EVALUATION PROCEDURE. WE PROVIDE THE CLASSIFICATION ACCURACY AND THE NUMBER OF FEATURES USED.

- Additive white noise resulting in 30, 20 and 10dB SNR.
- Pitch down and upconversion by 10 and 20% of pitch.
- Random signal cropping by 5, 10 and 15% of length.
- Telephone filtering with a [300, 3400]Hz bandpass filter.

These distortions are applied one at a time to each sound clip. Modified samples are then used as queries to the database (minus the original nondistorted sample) which allows comparing classification accuracies after distortion. We use in these tests only combinations of the best feature sets obtained in the classification task with normal quality audio. Results of the robustness analysis are synthesized in Table VI. We can see here that HV-MOTS consistently outperforms other approaches for cropping, pitch modification

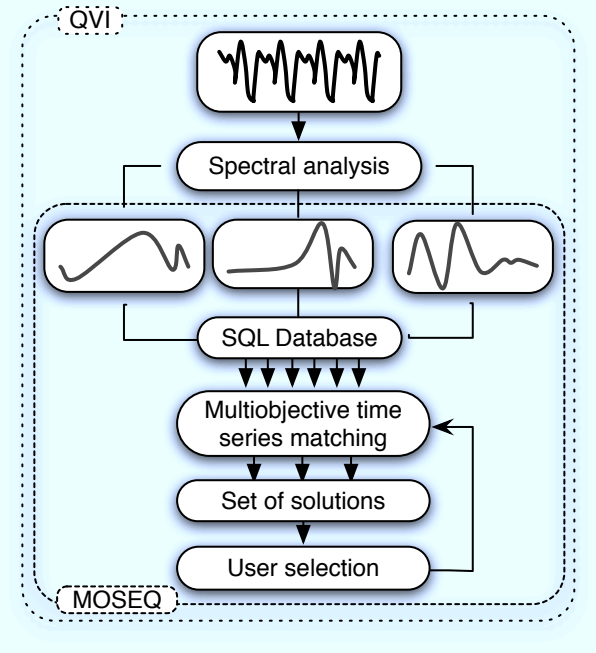

Figure 14. Algorithmic framework for two types of interaction. In *MOSEQ*, a set of time-evolving properties is drawn. The *MOTS* algorithm allows to find the set of efficient solutions. In *QVI*, the user can use his voice to perform an imitation of the desired properties (obtained from spectral analysis)

and telephone filtering and appears to be robust for these transformations. However, it seems that both multiobjective approaches are more affected by noise than mono-objective selections.

V. Future work and applications

We discuss the usability of our proposal in a QBE context and how its flexibility can potentially provide innovative paradigms for audio querying. Figure 14 summarizes the algorithmic framework for both applications. We start by showing in Section V-A how query results are presented to users in a QBE context. Then, we introduce two new interaction paradigms for audio retrieval. We present in Section V-B the *MultiObjective Spectral Evolution Query (MOSEQ)* where users can draw multiple temporal shapes required for a sample. We present in Section V-C a QBH-like system for audio clips called the *Query by Vocal Imitation (QVI)*. It allows users to imitate the evolution of spectral properties.

A. QBE results and representation

Figure 15 illustrates the results of two queries on the MuscleFish dataset. The first (left) is performed using a restaurant scene from the *crowds* class. The second (right) is performed using a sample of *female speech*. We present the results of both methods given the same set of features. Mono-objective selection provides an ordered list of results. However, there is no informed knowledge about how these choices were made whatsoever. Even with multiple dimensions involved, the results only offers an "*optimization line*" of fitness. Oppositely, the MOTS framework allows to obtain the complete optimization space. This representation informs the user on how solutions optimize various objectives. It also allows users to explore this space by focusing more on one objective than the other. If we look more closely at the results of these queries, we can see that the sets provided by the MOTS approach are more similar to the initial example query. In the first case, it appears that relevant results are spread over the criteria space which is revealed by the multiobjective optimization. On the other hand, mono-objective selection seems to get stuck on solutions performing averagely in both objectives. Hence, the representation provided by the MOTS approach also entails the cases where users seek parts of the query but not exactly the same content.

B. MultiObjective Spectral Evolution Query (MOSEQ)

The idea behind this interaction paradigm is that users create a mental representation of the temporal evolution of several spectral properties prior to the search. Therefore, the MOSEQ system would allow the user to select a set of features that are relevant to his query. Then, for each, he could simply draw their desired time series. This set acts as the target for this system, therefore, bypassing the need for a specific example. In a QBE context, the target $\mathcal{T}$ is the sound example for which similar instances have to be found. For the MOSEQ system, the target is represented by a set of time series features $\left\{\mathcal{F}^1\left(\mathcal{T}\right), ..., \mathcal{F}^K\left(\mathcal{T}\right)\right\}$ drawn by the user. Given this target and a sound sample $\mathcal{S}$, the k^{th} similarity function is the real-valued function $\mathcal{D}^k_{\mathcal{T}}\left(\mathcal{S}\right)$ that returns the distance between $\mathcal{S}$ and $\mathcal{T}$ along the k^{th} feature, i.e. between $\mathcal{F}^k(\mathcal{S})$ and $\mathcal{F}^k(\mathcal{T})$. It should be noted that it is possible to define a different similarity measure for each objective. By using the MOTS approach, the system could present the multidimensional space of audio clips. This allows to project time-evolving sound ideas and cope with the multidimensionality of timbre perception.

C. Query by Vocal Imitation (QVI)

The most straightforward way to communicate an idea is to use our voice. Most people have in some occasions imitated everyday sounds by using their voice and tried to match the temporal evolution of acoustic properties. Even with the inherent limitations of human voices, such as the *tessitura,* we can control remarkably specific sound qualities like the position of the formant frequencies, the type of phonation or the singer's formant [73]. We thus benefit from the high degree of expression of the singing voice, principally described by loudness, fundamental frequency and spectral envelope, which all vary dynamically with time. Sung imitations may convey valuable information as Pressing [58] indicates: "*One important resource in designing such expressivity is to use one's own voice to sing the expression in the part. Even if the sound quality is beyond the powers of your (or perhaps anyone's) voice, its time shaping may be imitable*". Despite the voice is limited in the range of timbres it can produce, much of vocal

	Normal	Pitch conversion				Cropping			Noise (SNR)			Telephone
		-20%	-10%	+10%	+20%	5%	10%	15%	10dB	20dB	30dB	
1-NN	91.69	88.02	90.71	89.73	86.06	90.95	90.46	90.46	**76.28**	76.28	81.66	90.95
5-NN	89.24	85.09	87.29	87.04	83.37	88.51	88.51	88.26	74.82	74.82	78.97	88.26
MOTS	85.82	76.53	83.37	84.60	78.97	84.60	84.11	84.11	66.26	66.26	74.08	84.11
HV-MOTS	**95.35**	**90.71**	**94.13**	**93.40**	**90.46**	**94.87**	**94.13**	**93.89**	74.82	**78.97**	**84.84**	**93.89**

Table VI
EFFECTS OF A SET OF DISTORTIONS ON CLASSIFICATION ACCURACY FOR DIFFERENT METHODS ON THE MUSCLEFISH DATASET.

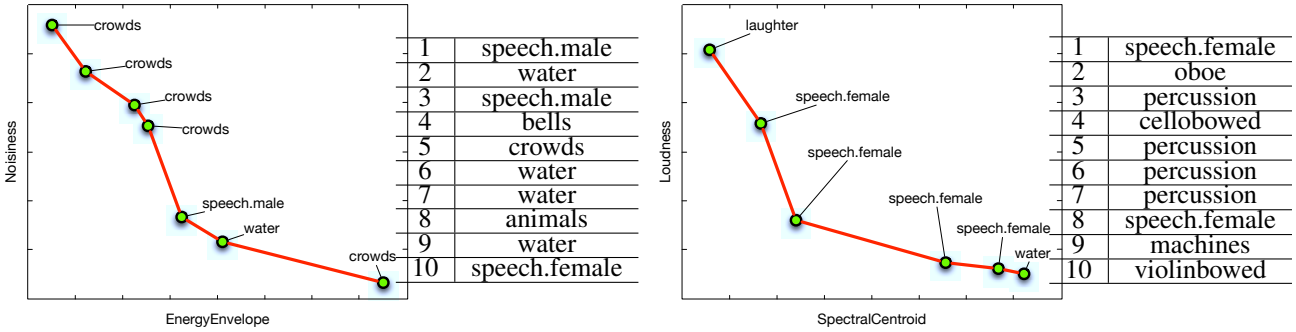

Figure 15. Comparison of different query results for multiobjective optimization and mono-objective selection in a QBE context. (Left) A sound taken from a restaurant scene and belonging to the *crowd* class. (Right) A clip taken from the *female speech* class.

expression can be captured, not in the absolute timbre but in the relative temporal change of timbre. Therefore, a natural way of querying sound samples would be to perform a vocal imitation based on spectral shapes. The QVI problem could be reduced to a MOSEQ problem. Furthermore, the lack of spectral and harmonic control can be circumvented by manually drawing some of the temporal shapes or *mapping* useful vocal descriptors to unrelated spectral features.

VI. CONCLUSIONS

We have presented in this paper a novel approach for content-based audio classification and retrieval. By analyzing sound samples with the full scope of spectral descriptors currently available, we are able to obtain a database with a precise knowledge on various audio properties. This database can allow queries based on the temporal shape of spectral properties by using time series analysis techniques. However, our goal was to go beyond this scheme and introduce innovative concepts of interaction. For performing queries on several time-evolving properties simultaneously, we proposed to merge time series analysis and multiobjective optimization in a common framework. We thus stated this new problem as *MultiObjective Time Series* (*MOTS*) matching. We evaluated this approach in a classification framework using two datasets. We showed that our approach outperforms the state-of-art methods on a reference dataset even with a limited number of features involved. Mean classification accuracies seems to indicate that the hypervolume multiobjective optimization retains the discriminative power of the best feature involved, whereas mono-objective selection is confined by the worst feature of the set. We also showed the robustness of our approach to several classes of distortions. We presented query results, which allow the user to see how solutions optimize the different objectives. For future work, we need to assess the usability of the QVI framework as an intuitive way to interact with sound samples through comprehensive user studies. An extremely useful investigation would also consist in analyzing which spectral descriptors can truly be controlled in these types of queries. Regarding the performance of the algorithm itself, we envisioned to test several representations and indexing methods in order to determine which among them provides the best efficiency. We believe that the generic approach of MOTS matching could be applied to a whole range of concrete problems including medical diagnosis, chemical engineering and genetic analysis. In fact, this new approach could be beneficial to several topics where multiobjective optimization has proven to be useful. Enhancing such approaches by allowing the use of time series data would provide more powerful and flexible analysis tools.

REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search In Sequence Databases," in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. Springer, 1993, pp. 69–84.

[2] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," in *Proceedings of the 21th International Conference on Very Large Data Bases*, San Francisco, USA, 1995, pp. 490–501.

[3] B. Bakshi and G. Stephanopoulos, "Representation of process trends–IV. Induction of real-time patterns from operating data for diagnosis and supervisory control," *Computers & Chemical Engineering*, vol. 18, no. 4, pp. 303–332, 1994.

[4] G. Ballet, R. Borghesi, P. Hoffmann, and F. Levy, "Studio online 3.0 : An internet "killer application" for remote access to ircam sounds and processing tools," in *Actes des Journees Informatique Musicale*, Paris, France, 1999.

[5] L. Barrington, A. Chan, D. Turnbull, and G. Lanckriet, "Audio information retrieval using semantic similarity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, vol. 2, 2007, pp. 722–725.

[6] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles," *ACM SIGMOD Record*, vol. 19, no. 2, pp. 322–331, 1990.

[7] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 1075–1082, 2009.

[8] R. Cai, L. Lu, and H. Zhang, "Using structure patterns of temporal and spectral feature in audio similarity measure," in *Proceedings of the 11th ACM international conference on Multimedia*, 2003, pp. 219–222.

[9] P. Cano and M. Koppenberger, "Automatic sound annotation," in *Proceedings of the 14th IEEE Workshop on Machine Learning for Signal Processing*. IEEE, 2004, pp. 391–400.

[10] P. Cano, M. Koppenberger, S. Le Groux, J. Ricard, P. Herrera, and N. Wack, "Nearest-neighbor generic sound classification with a wordnet-based taxonomy," in *Proceedings of the 116th AES Convention*. Berlin, Germany: Citeseer, 2004.

[11] M. Casey, "General sound classification and similarity in mpeg-7," *Organised Sound*, vol. 6, no. 02, pp. 153–164, 2001.

[12] ——, "Reduced-rank spectra and minimum-entropy priors as consistent and reliable cues for generalized sound recognition," in *Workshop for Consistent & Reliable Acoustic Cues*, 2001.

[13] ——, "Sound classification and similarity," *Introduction to MPEG-7: Multimedia Content Description Interface*, pp. 309–317, 2002.

[14] ——, "Acoustic lexemes for organizing internet audio," *Contemporary Music Review*, vol. 24, no. 6, pp. 489–508, 2005.

[15] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[16] K. Chan and A. Fu, "Efficient time series matching by wavelets," in *Proceedings of the 15th IEEE International conference on data engineering*, Sydney, Australia, 1999, pp. 126 – 133.

[17] W. Conover, "Practical nonparametric statistics," 1980.

[18] G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," *Principles of Data Mining and Knowledge Discovery*, pp. 88–100, 1997.

[19] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[20] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[21] J. Downie, "The scientific evaluation of music information retrieval systems: Foundations and future," *Computer Music Journal*, vol. 28, no. 2, pp. 12–23, 2004.

[22] F. Edgeworth, "Mathematical psychics," *History of Economic Thought Books*, 1881.

[23] P. Esling and C. Agon, "Time series data mining," *ACM Computing Surveys*, vol. 45, no. 1, 2012 (to appear).

[24] C. Faloutsos, M. Ranganathan, and Y. Manolopulos, "Fast subsequence matching in time-series databases," *SIGMOD Record*, vol. 23, pp. 419 – 429, 1994.

[25] C. Fonseca, L. Paquete, and M. López-Ibánez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *IEEE Congress on Evolutionary Computation (CEC'2006)*, 2006, pp. 1157–1163.

[26] T. Friedrich, C. Horoba, and F. Neumann, "Multiplicative approximations and the hypervolume indicator," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 2009, pp. 571–578.

[27] A. Ghias, J. Logan, D. Chamberlin, and B. Smith, "Query by humming: musical information retrieval in an audio database," in *Proceedings of the third ACM international conference on Multimedia*. ACM, 1995, pp. 231–236.

[28] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database : music genre database and musical instrument sound database," in *Proceedings of the 4th International Conference on Music Information Retrieval*, Washington, USA, 2003, pp. 229 – 230.

[29] J. Grey, "Multidimensional perceptual scaling of musical timbres," *Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977.

[30] S. Gudmundsson, T. Runarsson, and S. Sigurdsson, "Support vector machines and dynamic time warping for time series," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 2772–2776.

[31] G. Guo and S. Li, "Content-based audio classification and retrieval by support vector machines," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 209–215, 2003.

[32] G. Guo, H. Zhang, and S. Li, "Boosting for content-based audio classification and retrieval: an evaluation," in *IEEE International Conference on Multimedia and Expo (ICME 2001)*, 2001, pp. 997–1000.

[33] M. Helén and T. Lahti, "Query by example methods for audio signals," in *Proceedings of the 7th Nordic Signal Processing Symposium*, 2006, pp. 302–305.

[34] M. Helen and T. Lahti, "Query by example in large databases using key-sample distance transformation and clustering," in *Proceedings of the 9th IEEE International Symposium on Multimedia Workshops (ISMW'07)*, 2007, pp. 303–308.

[35] M. Helen and T. Virtanen, "Query by example of audio signals using euclidean distance between gaussian mixture models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2007, pp. 225–228.

[36] K. Islam, K. Hasan, Y. Lee, and S. Lee, "Enhanced 1-nn time series classification using badness of records," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM, 2008, pp. 108–113.

[37] J. Jang, C. Hsu, and H. Lee, "Continuous HMM and Its Enhancement for Singing/Humming Query Retrieval." ISMIR 2005, 6th International Conference on Music Information Retrieval, 2005.

[38] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

[39] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[40] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[41] L. Latecki, R. Lakaemper, and D. Wolter, "Optimal partial shape similarity," *Image and Vision Computing*, vol. 23, no. 2, pp. 227–236, 2005.

[42] M. Lesaffre, M. Leman, K. Tanghe, B. De Baets, H. De Meyer, and J. Martens, "User-dependent taxonomy of musical features as a conceptual framework for musical audio-mining technology," in *Proc. of the Stockholm Music Acoustics Conference*, 2003.

[43] G. Li and A. Khokhar, "Content-based indexing and retrieval of audio data using wavelets," in *IEEE International Conference on Multimedia and Expo*, vol. 2. IEEE, 2000, pp. 885–888.

[44] S. Li, "Content-based audio classification and retrieval using the nearest feature line method," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 5, pp. 619–625, 2000.

[45] Y. Li and Y. Hou, "Search audio data with the wavelet pyramidal algorithm," *Information processing letters*, vol. 91, no. 1, pp. 49–55, 2004.

[46] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM New York, NY, USA, 2003, pp. 2–11.

[47] J. Lin, E. Keogh, S. Lonardi, J. Lankford, and D. Nystrom, "Visually mining and monitoring massive time series," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 460–469.

[48] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[49] D. Mazzoni and R. Dannenberg, "Melody matching directly from audio," in *2nd Annual International Symposium on Music Information Retrieval*, 2001, pp. 17–18.

[50] S. McAdams, "Psychological constraints on form-bearing dimensions in music," *Contemporary Music Review*, vol. 4, no. 1, pp. 181–198, 1989.

[51] S. McAdams, S. Winsberg, S. Donnadieu, G. Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common

dimensions, specificities, and latent subject classes," *Psychological research*, vol. 58, no. 3, pp. 177–192, 1995.

[52] I. Mierswa and M. Wurst, "Efficient case based feature construction," *Machine Learning: ECML 2005*, pp. 641–648, 2005.

[53] J. Miller and E. Carterette, "Perceptual space for musical structures," *The Journal of the Acoustical Society of America*, vol. 58, p. 711, 1975.

[54] V. Pareto, "Cours d'Economie Politique, volume I and II," *F. Rouge, Lausanne*, vol. 250, 1896.

[55] S. Pauws, "CubyHum: a fully operational query by humming system," in *Proceedings of ISMIR*, 2002, pp. 187–196.

[56] G. Peeters, "A large set of audio features for sound description in the cuidado project," IRCAM, Paris, Tech. Rep., 2004.

[57] G. Peeters and E. Deruty, "Automatic morphological description of sounds," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3801, 2008.

[58] J. Pressing, *Synthesizer performance and real-time techniques*. AR Editions, Inc. Madison, WI, USA, 1992.

[59] H. Qi, P. Hartono, K. Suzuki, and S. Hashimoto, "Sound database retrieved by sound," *Acoustical Science and Technology*, vol. 23, no. 6, pp. 293–300, 2002.

[60] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Time-series classification in many intrinsic dimensions," in *10th SIAM International Conference on Data Mining*. Citeseer, 2010.

[61] D. Rafiei and A. Mendelzon, "Efficient Retrieval of Similar Time Sequences Using DFT," in *Proceedings of the 5th Int. Conf. of Foundations of Data Organization and Algorithms*, 1998, pp. 249–257.

[62] C. Ratanamahatana, E. Keogh, A. Bagnall, and S. Lonardi, "A novel bit level time series representation with implication of similarity search and clustering," *Advances in Knowledge Discovery and Data Mining*, pp. 771–777, 2005.

[63] M. Reyes-Gomez and D. Ellis, "Selection, parameter estimation, and discriminative training of hidden markov models for general audio modeling," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 1, 2003, pp. 70–73.

[64] S. Samson, R. Zatorre, and J. Ramsay, "Multidimensional scaling of synthetic musical timbre: Perception of spectral and temporal characteristics," *Canadian Journal of Experimental Psychology*, vol. 51, no. 4, pp. 307–315, 1997.

[65] X. Shao, C. Xu, and M. Kankanhalli, "Applying neural network on the content-based audio classification," in *Proceedings of the 4th IEEE Joint International Conference on Information, Communications and Signal Processing*, vol. 3, 2003, pp. 1821–1825.

[66] H. Shatkay and S. Zdonik, "Approximate queries and representations for large data sequences," in *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, 1996, pp. 536–545.

[67] J. Shieh and E. Keogh, "isax : indexing and mining terabyte sized time series," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 623–631.

[68] M. Slaney, "Mixtures of probability experts for audio retrieval and indexing," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 1. IEEE, 2002, pp. 345–348.

[69] ——, "Semantic-audio retrieval," in *IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2002, p. 4108.

[70] S. Subramanya, R. Simha, B. Narahari, and A. Youssef, "Transform-based indexing of audio data for multimedia databases," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems' 97*. IEEE, 1997, pp. 211–218.

[71] S. Subramanya and A. Youssef, "Wavelet-based indexing of audio data in audio/multimedia databases," in *Multi-Media Database Management Systems, 1998. Proceedings. International Workshop on*. IEEE, 1998, pp. 46–53.

[72] S. Sundaram and S. Narayanan, "Classification of sound clips by two schemes: using onomatopoeia and semantic labels," in *Multimedia and Expo, 2008 IEEE International Conference on*. IEEE, 2008, pp. 1341–1344.

[73] J. Sundberg, "Level and Center Frequency of the Singer's Formant* 1," *Journal of voice*, vol. 15, no. 2, pp. 176–186, 2001.

[74] A. Uitdenbogerd and J. Zobel, "Melodic matching techniques for large music databases," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. ACM, 1999, p. 66.

[75] T. Virtanen and M. Helén, "Probabilistic model based similarity measures for audio query-by-example," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, 2007, pp. 82–85.

[76] C. Wan, M. Liu, and L. Wang, "Content-based sound retrieval for web application," *Web Intelligence Research*, pp. 389–393, 2001.

[77] G. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[78] D. Wessel, "Timbre space as a musical control structure," *Computer music journal*, vol. 3, no. 2, pp. 45–52, 1979.

[79] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *Multimedia, IEEE*, vol. 3, no. 3, pp. 27–36, 1996.

[80] J. Xue, G. Wichern, H. Thornburg, and A. Spanias, "Fast query by example of environmental sounds via robust and efficient cluster-based indexing," in *Acoustics, Speech and Signal Processing. ICASSP 2008. IEEE International Conference on*, 2008, pp. 5–8.

[81] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81–93, 1999.

[82] T. Zhang and C. Kuo, "Classification and retrieval of sound effects in audiovisual data management," in *Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on*, vol. 1. IEEE, 1999, pp. 730–734.

[83] ——, "Hierarchical classification of audio data for archiving and retrieving," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, 1999, pp. 3001–3004.

[84] Y. Zhu, M. Kankanhalli, and C. Xu, "Pitch tracking and melody slope matching for song retrieval," *Advances in Multimedia Information Processing (PCM 2001)*, pp. 530–537, 2001.

[85] Y. Zhu and D. Shasha, "Query by humming: a time series database approach," in *Proc. of SIGMOD*, 2003, p. 675.

[86] ——, "Warping indexes with envelope transforms for query by humming," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003, pp. 181–192.

[87] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.

[88] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 862–876.