Audio Engineering Society

Convention Paper

Presented at the 132nd Convention
2012 April 26–29   Budapest, Hungary

Automated Horizontal Orchestration based on Multichannel Musical Recordings

Maximos A. Kaliakatsos–Papakostas[1], Andreas Floros[2], and Michael N. Vrahatis[1]

[1] *Computational Intelligence Laboratory, (CILab), Department of Mathematics, University of Patras, GR-26110 Patras, Greece*

[2] *Department of Audio and Visual Arts, Ionian University, GR-49100 Corfu, Greece*

Correspondence should be addressed to Maximos A. Kaliakatsos–Papakostas (maxk@math.upatras.gr)

ABSTRACT

Orchestration of computer-aided music composition aims to approximate musical expression using vertical instrument sound combinations, i.e. through finding appropriate sets of instruments to replicate synthesized sound samples. In this work, we focus on horizontal orchestration replication, i.e. the potential of replicating the instantaneous intensity variation of a number of instruments that comprise an existing, target music recording. A method that efficiently performs horizontal orchestration replication is provided, based on the calculation of the instrumental Intensity Variation Curves. It is shown that this approach achieves perceptually accurate automated orchestration replication when combined with automated music generation algorithms.

1. INTRODUCTION

Music composition is a field of human creativity that computer science cannot yet penetrate. The sonic realization of a musical composition is accomplished through assigning certain tonal roles to musical instruments or abstract sound generators throughout a musical piece. The transition from symbolic music to the sonic domain is performed through orchestration, which discusses the utilization of certain timbres and their intensities at a specific time instance. Until recently, the orchestration of automatically composed musical pieces had not been a subject of thorough study. Specifically, the intensity of the instruments that compose the generated piece was provided in terms of probabilistic or algebraic functions [17], evolutionary processes or even added manually [7].

Recent research approaches towards automatic orchestration have yielded important results. The main goal of such existing works is to exploit information provided by spectral analysis of short sound signals in order to produce similar signals with the use of musical instruments. Some of these works resulted into the implementation of systems that perform these spectral replication tasks, like *Orchidée* [1] and *SPORCH* [10].

In this work, we examine a different automated orchestration perspective that aims to integrate computer–aided composition systems with the utilization of certain musical instrument dynamics variation templates obtained from existing multichannel recordings. Hence, we mainly focus on extracting information related to the time–dependent intensity variation of individual instruments comprising typical musical recordings. By extracting information about the intensities of instruments that constitute a target recording, we are able to algorithmically compose novel pieces that have similar instrumental structure. Furthermore, indications are provided that the examined approach provides novel recordings that are perceptually more similar to the target recording, than recordings with the same instrumental structure but random intensity variations.

The paper at hand is organized as follows. In Section 2 we analyze in detail the motivation and the aims of this work. In Section 3, we describe the problem of *horizontal orchestration replication* and further discuss a framework of methods capable of replicating the orchestration of a recorded musical piece. Section 4 proposes an evaluation procedure for the above methods, while Section 5 provides an algorithm that detects the intensity variations of instruments throughout a multichannel music track. The latter algorithm is tested and evaluated in Section 6. Finally, conclusions and pointers for future work are discussed in Section 7.

2. MOTIVATION AND AIMS

Recent works considered orchestration replication in the frequency domain. These works aim at providing to the (automatic or even human) music composers the knowledge of how to replicate specific timbre patterns appeared within a target musical track, with the use of a specified set of instrumental recordings [1, 10, 11, 2]. The aforementioned approaches are applied in the *vertical* instrument domain [2], where a specific combination of instruments is directly mapped to an existing orchestrational timbre being static for a short period of time. A shortcoming of these approaches concerns their incompatibility with Automatic Algorithmic Composition Systems (AACSs), since they aim at providing a set of sonic-musical components (e.g. tones and durations) that would most likely violate the inherent independence of the AACS.

Our aim here is to develop an orchestration technique that acts in the *horizontal* instrument domain. This technique should identify the instantaneous intensity level of the instruments that take part in a specific musical recording as a function of time and apply it in the context of orchestration replication combined with algorithmically controlled music synthesis methods. Under this perspective, horizontal orchestration replication (HOR) can be combined with an AAC (e.g. [13, 9]), with the use of certain instruments that have similar acoustical roles to those identified in the originally recorded target piece. Since no framework currently exists for methods that extract knowledge from the horizontal orchestration of a recorded piece, particular aims of this work are to:

1. define the problem of *horizontal orchestration replication* and provide a general outline of realization methods
2. deploy an evaluation scheme for horizontal orchestration replication efficiency assesment and
3. propose a HOR method and evaluate it.

3. OUTLINE OF METHODS FOR HORIZONTAL ORCHESTRATION REPLICATION

3.1. Detailed HOR framework definition

Suppose that we have a recorded musical piece the orchestration of which we wish to replicate. In the next paragraphs, this will be termed as *target recording*. Firstly, we have to define which aspects of its orchestration we are referring to. It is well–known that the notion of orchestration covers a wide range of musical attributes, from timbres and intensities of musical instruments to rhythmical and tonal tension of melodic structure [14]. Since we focus in the context of automated music composition, the melodic properties have already been defined through systems that automatically generate specific rhythmical and tonal sequences with desired structures.

By orchestration replication of a target recording we refer to the aspects of its music composition that are complementary to its melodic structure, thus the timbre and the intensities of its arranging instruments. More precisely, we could say that *orchestration replication based on a target recording is the production of a new music piece that inherits similar instrumental structure both in terms of timbre and instantaneous instrument intensity level.* If the instruments that take part into this music waveform are already known (which typically represents the case of a multichannel master recording), the information required to carry out the orchestration task is obviously the instruments' intensity levels as a function of time[1]. We hereby refer to this problem as *horizontal orchestration replication.*

A potential employment of horizontal orchestration based on multichannel target recording is defined within the context of real–time, algorithmically controlled music synthesis. For example, in this case, a set of instruments could be defined and the melody of each instrument could be instantly generated by an automatic music generation system. However, in this case the intensity of each instrument at any time instance must be additionally defined. This could be done using a method that replicates the orchestration of an already recorded musical piece, that is considered to be a suitable orchestration template, taking into account several parameters, such as the music genre for example.

3.2. General method description

Based on the previous analysis, a general orchestration replication method should be able to a) produce an orchestration template based on a target recording and b) capture the timbre and intensity characteristics of the instruments that participate in it. These requirements can be fulfilled by a method that performs two tasks:

Task 1: separate the instruments that exist in the target recording as separate sources and

Task 2: for every separated instrument, define its intensity level as a function of short time intervals throughout the target recording.

Task 1 can be accomplished by any blind source separation algorithm, taking into account potential spectral overlapping issues frequently appeared in typical audio/music recordings. Such algorithms have been extensively studied in the literature [5, 16] and ideally extract waveforms for each instrument from a target (i.e. stereo) recording. Moreover, provided that audio master recordings are usually available in multichannel formats, with each channel representing an instrumental source signal, this task can be omitted. The timbre of each instrument could be predefined, for example, if we wish to replicate the orchestration of a string quartet, we may assume that the instruments in the target composition will be a cello, two violas and a violin. In this way, the separation of the full spectrum of each instrument in the target recording is not necessary.

In **Task 2**, the instantaneous intensity level of each instrumental sound source has to be defined throughout the piece. These intensities can be represented as time-domain curves, termed here as Intensity Variation Curves (IVCs). The value $v = \mathrm{IVC}(t)$ of an IVC that corresponds to a specific instrument, is the intensity level of this instrument at a time instant equal to t. Time and instrument intensity level can be expressed in terms of any desired units (i.e. seconds and decibels, or meter subdivisions and MIDI velocity respectively).

The vast research stream towards audio source separation has yielded impressive results so far. Evaluation and comparison of these methods are beyond the scope of this work. In this paper we only consider multichannel target recordings (or equivalently we assume that audio separation is performed ideally) and propose a scheme for examining whether the IVCs produced by an algorithm are *consistent.* Furthermore, we propose a method for the IVC extraction of instruments separated from a target recording and test its consistency.

[1] The majority of musical instruments can produce multiple heterogeneous timbres, i.e. pizzicato and glissando in a violin. AACs though, treat such expression differences of the same instrument as belonging to separate instruments. Thus the expression potential of instruments is not considered crucial information for this work.

4. METHODOLOGY FOR EVALUATING AN IVC EXTRACTION ALGORITHM

In this section we propose a scheme for evaluating an IVC extraction algorithm. Thus, we deploy a criterion that judges whether an algorithm that performs **Task 2** above produces satisfactory results. This evaluation scheme will be more effective if we consider that instrument separation has been performed ideally by an algorithm that performs **Task 1**. However, as mentioned previously, we hereby consider multichannel recorded tracks. Hence, for the purposes of this work, **Task 1** can be ignored.

In a typical music piece recording, each instrument waveform is recorded by a musician or synthesized using computer aided means. In both cases, an “intensity level plan” is applied, defining the instantaneous energy distribution that corresponds to the specific instrument. This intensity level plan strongly depends on the musicians personal style of playing, could be dictated by a conductor or could be seeded as input to the automatic performance system. An algorithm that extracts the IVC of an instrument, should produce an IVC similar to the targeted “intensity level plan” of the performing musician or computer.

However, in practice, one cannot be sure about the exact intensity level plan that a musician utilizes while performing. On the other hand, an AAC-based performance system can utilize a predefined intensity plan throughout the recording of an instrument. This computer-aided “intensity level plan” can be represented by any curve, using the MIDI velocity protocol with values between 0 and 127. An algorithm that extracts the IVC of the aforementioned computer synthesis can be considered to be *consistent*, if it produces an IVC similar to the initial MIDI “intensity level plan”.

An automatic music composition and performance system should be utilized to best-match the aforementioned recording in terms of dynamics throughout its entire duration. To simulate realistic human performance this composition and performance system should be able to produce a) diverse rhythmical patterns and b) accentuation intensity variations that in some degree violate the intensity plan. It is important that these two performance attributes will be taken under consideration in order to avoid possible misinterpretations of the algorithm utilization on human performance recordings.

Furthermore, it is crucial that more than one intensity level plans and musical instruments should be recorded, since a method has to be effective with all possible types of instrumental properties. Thus, the method should produce the desirable results in terms of varying timbres and attack decays. Taking into account these considerations, an IVC extraction algorithm evaluation methodology can be analytically described by the following three steps:

Step 1: Given a set of “intensity plan” curves, record several musical instruments produced by an automatic composition and performance system, using several rhythmical and accentuation patterns.

Step 2: Extract the IVCs of these recordings using the method under evaluation.

Step 3: If the difference between the derived IVCs and the intensity plan curves is below a predefined threshold, then this algorithm is characterized as consistent.

Furthermore, the mixed recording derived by the aforementioned IVCs should exhibit similar orchestration characteristics with the target one. The similarity measures that may be considered however, should be tolerant with tonal and rhythmical characteristics, since these features are controlled by the AACS. Among the audio features that capture such characteristics are the Mel–Frequency Cepstral Coefficients (MFCCs) [6] and the Bark scale subdivision of the audible frequency range [18]. Additionally, we are able to compute the difference of the total loudness between the target and the replicated music signals in short consecutive time intervals using the Stevens loudness method [3]. A detailed description of the features we employ is provided in Section 6.

5. IVC EXTRACTION ALGORITHM

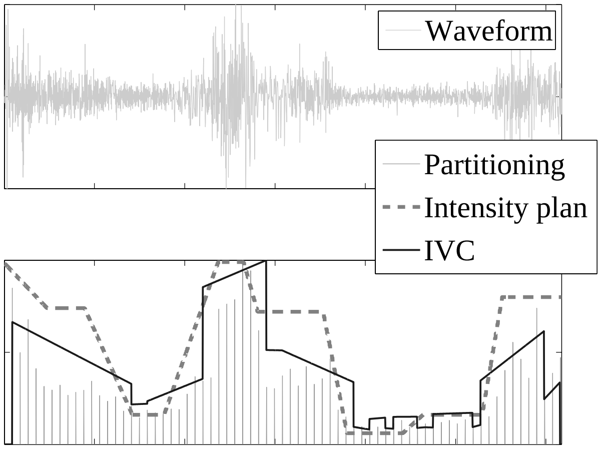
In this section we propose an algorithm that performs **Task 2**, as defined in Section 3, aiming to extract the IVC for each instrument of a multichannel recording.

Before we move on with the description of the algorithm, we need to demonstrate the notation that will be used. Consider two vectors, $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$. If we perform the *Linear Least Squares* (LLS) algorithm over the n points on the xy plane, $(\vec{x}, \vec{y}) = (x_1, y_1), \ldots, (x_n, y_n)$, we obtain a line on this plane, $y = gx + c$, where g is its gradient and c its elevation constant. We denote the LLS algorithm as a function with input the n points and output the constants g and c of the regression line, that is $[g, c] = \mathrm{LLS}(\vec{x}, \vec{y})$. Then, if we have two sets of symbols A and B, the concatenated set of symbols C is denoted by $C = [A, B]$. Finally, if A is a set of numbers, then $\mu_A = \mathrm{mean}(A)$ is the mean value of A and $\sigma_A = \mathrm{std}(A)$ is the standard deviation of A.

Returning back to Algorithm 1 description, we consider a recorded musical instrument, the waveform of which we denote by X and we wish to extract its IVC. We consider a *partition* $P = [s_1, s_2, s_3, ..., s_n,]$, of this recording in equally spaced time segments, $t_0, t_1, t_2, ...t_n$ of length t[2]. Segment s_i begins at time t_{i-1} and ends at time t_i. Within each segment s_i, we calculate the mean energy $E(s_i)$ of the sound wave X. In Figure 1 a recorded instrument waveform and the mean energies within segments of an 1 second partition are illustrated. The lower graph also depicts the intensity level plan and the IVC obtained by the proposed Algorithm 1.

The main concept of the above Algorithm is to create a curve that groups partitions sharing common monotonic mean energy behavior. Thus the algorithm should acknowledge whether a series of consecutive segments are imposing a crescendo, a decrescendo or a dynamically steady part. Furthermore, the intensity level of a steady part and the beginning and ending intensities of crescendo and decrescendo parts should be correctly defined. There is a case though, where consecutive segments demonstrate great mean energy differences while belonging to a part of the piece with the same intensity plan. This could happen for example, in segments between an intense note and a pause, within an intense part. The following paragraph describes a way to tackle this problem, which is resolved in lines $9 - 11$ of Algorithm 1 description provided below.

Fig. 1: The waveform of a recording of an instrument, the mean energy values for a partition of 1 second together with the intensity level plan and the resulting IVC.



The algorithm considers two consecutive pairs of segments, $A = [s_{i-1}, s_i]$ and $B = [s_i, s_{i+1}]$, and their respective mean energy values, $E(A) = [E(s_{i-1}), E(s_i)]$ and $E(B) = [E(s_i), E(s_{i+1})]$. Linear Least Squares (LLS) regression is performed for both $(A, E(A))$ and $(B, E(B))$, and the output is a set of two lines with two probably different gradients and elevation constants. If the gradients differ above a tolerance level, then they are considered to contain a transition between a note and a silence, or between notes of extremely different intensities due to accentuation. To avoid unsafe results through pairs of segments with great gradient difference, we consider such segments as belonging to the same intensity scope. In this case, pairs of segments A and B are concatenated and named as $A = [s_{i-1}, s_{i+1}]$. LLS is performed on the new concatenated A, and B propagates to the next pair of segments, so that $B = [s_{i+1}, s_{i+2}]$. If the gradient difference is less than the tolerance level, both pairs propagate to the next, thus $A = [s_i, s_{i+1}]$ and $B = [s_{i+1}, s_{i+2}]$.

For proper values of t, k, m and s, Algorithm 1 is expected to produce an IVC that matches the "intensity plan" curve. The gradient multiplier, k, is used for the algorithm to distinguish whether or not a new LLS line is needed for the forthcoming segment. The elevation adjustment multipliers, m and s, are

[2] If t is not a divisor of the total length of the recording, we drop off the final spare segment of smaller length.

Algorithm 1 IVC extraction algorithm

```
Require: waveform $X$, segment time length $t$, gradient multiplier $k$, elevation adjustment multipliers, $m$ and $s$
Ensure: The IVC of $X$ as a function of time, $y = \text{IVC}(x)$
Create a partition of $X$ with segments of the same length (except the last segment probably) $t$ seconds. We then have a partition $P = [s_1, s_2, s_3, ..., s_n,]$, with each segment $s_i$ covering an area between time instances $t_{i-1}$ and $t_i$, with $t_i - t_{i-1} = t$.
Compute the mean gradient between consecutive pairs of segments, $\mu_X$, throughout the piece $X$.
$A \leftarrow [s_1, s_2]$
$E(A) \leftarrow [E(s_1), E(s_2)]$
for $i = 3$ to $n$ do
    $[g_A, c_A] = \text{LLS}(A, E(A))$
    $\mu_{E(A)} \leftarrow \text{mean}(E(A))$
    $\sigma_{E(A)} \leftarrow \text{std}(E(A))$
    $\text{IVC}(A) = g_A\ x + c_A + m\ \mu_{E(A)} + s\ \sigma_{E(A)}$
    $B \leftarrow [s_{i-1}, s_i]$
    $[g_B, c_B] \leftarrow \text{LLS}(B, E(B))$
    $\text{IVC}(B) = g_B\ x + c_B + m\ \mu_{E(B)} + s\ \sigma_{E(B)}$
    if $|g_B - g_A| > k\ \mu_X$ then
        $A \leftarrow [A, B]$
    else
        $A \leftarrow B$
    end if
end for
```

used for fine-fitting the IVC on the corresponding intensity plan. The described algorithm uses these multipliers on mean value and standard deviation of energies in a segment.

The derived IVC is a set of linear segments. Piecewise linear models have been studied previously in the literature [15]. In this algorithm though, linear segments are adapted to the mean value and standard deviation of energy in the respective segments of the waveform. For this, we call the algorithm under discussion, *Self-Adaptive Piecewise Linear Least Squares*.

6. EXPERIMENTS AND RESULTS

We first demonstrate the procedure that we followed to tune the parameters t, k, m and s of the proposed IVC extraction algorithm. These parameters are tuned separately for the 18 musical instruments presented in Table 1, yielding 18 different combinations that provide optimally fitted IVC curves for each instrument. Sampled sounds of these instruments were used, which are included in Ableton Live additional libraries. These instruments are separated in 3 categories, grouped according to timbre. These categories are *melodic*, *polyphonic* and *bass*. During the recording of an instrument belonging to the melodic or the bass categories, the AACS was composing in monophonic mode, allowing only one note at any triggered onset. On the other hand, for the polyphonic instruments 1 to 4, multiple notes were allowed to be played simultaneously.

Based on the aforementioned parameters, we applied the proposed approach on 30 recordings of automatically composed musical content, created by a system developed by the authors using MAX/MSP. This system was able to produce diverse rhythmical patterns and accentuation intensity variations, as discussed in Section 4. Different combinations of the 18 aforementioned musical instruments were used for these recordings and each instrument was recorded in a single monophonic track. A sample piece can be found in [4], while the complete set of pieces, together with the separate tracks for each instrument are available upon request.

6.1. Tuning the parameters

For the algorithm described in Section 5 the parameters t, k, m and s have to be defined. We utilized the Differential Evolution (DE) algorithm [12] to tune these parameters' values, using the recordings of the 18 instruments, for which the intensity plans were known. These intensity plans covered the full dynamic range of the instruments and included at least 10 seconds of performance with steady dynamics, gradual changes of intensity (crescendo and decrescendo) and sudden intensity changes. Four recordings with 70 seconds duration of automatically composed performances were also used to tune each instrument's parameters with different playing styles in terms of speed and accentuation intensities.

The DE algorithm was used for minimizing an objective function with respect to some variables. In this case the objective function under optimization was the sum of Euclidian distances of the respective intensity plans and produced IVCs for all instruments on their respective tuning recordings. The optimal values for each instrument are shown in Table 1.

Table 1: The 18 instruments used for the experimental results and their optimal parameters.

	instrument	t	k	m	s
melodic	Fr. horn	1.03	0.76	1.12	-0.96
	En. horn	0.85	0.99	0.11	0.74
	Oboe	0.99	0.69	0.74	0.26
	Trumpet	0.99	0.85	0.18	1.18
	Viola	0.54	0.80	0.94	-0.55
	Violin	1.12	0.77	1.02	-0.16
polyphonic	Cl. guitar	1.13	0.97	1.25	0.20
	Harp	0.96	0.85	1.99	-1.94
	El. guitar	0.81	0.77	-0.25	1.37
	El. piano	1.29	0.82	0.45	0.30
	Grand piano	0.63	0.39	0.41	0.66
	Xylophone	0.61	0.09	-0.73	1.70
bass	El. Bass	0.30	0.39	0.22	1.49
	Ac. bass	0.28	0.37	0.31	1.55
	Tuba	1.31	0.78	1.75	-0.78
	Bass trombone	0.52	0.71	0.44	0.50
	Cello	0.50	0.58	1.20	-0.76
	Double bass	0.66	0.75	0.25	0.76

6.2. HOR evaluation

For the evaluation of our approach we have used three sets of recordings with music content that was automatically composed. These sets are denoted as T_i, R_i and X_i, $i \in \{1, 2, \ldots, 30\}$, and all these sets comprise of 30 recordings each, with 70 seconds duration. The 30 recordings belonging to the T_i set are called the *target* recordings and the instrumentation of each T_i is a combination of 3 instruments, one from each category (melodic, polyphonic and bass). The combinations were randomly created taking into account that no instrumentation should occur more than once, and that all instruments were used exactly 5 times. The initial intensity plans for the T_i recordings were random combinations of steady parts and sudden and gradual intensity changes. Each instrument for all T_i recordings was recorded as a separated track.

Based on the multitrack recordings provided by each T_i, we used the IVC extraction algorithm to replicate their orchestration. We have created the set R_i which comprise the set of orchestration replications of T_i. Specifically, each recording R_i is the orchestration replicate of T_i, with $i \in \{1, 2, \ldots, 30\}$, in terms of instrumentation and instrument intensities. Thus, the respective recordings in R_i and T_i are recorded with the same combination of instruments and the intensity variation of each instrument in R_i is a replicate of the intensity variation of the respective instrument in T_i, which is extracted by the proposed IVC extraction algorithm.

The X_i set includes recordings that have similar instrumentation with the T_i and R_i recordings, but the intensity variation of the instruments in each X_i was random. The system that controlled the intensity variations of X_i is the same system that we used for the T_i recordings, but with different random number seed. Thus, the intensities of instruments in X_i were different to the intensities of the respective instruments in T_i. The automatic composition system that composed music in all recordings was set in a relatively "steady" composition mode, to allow the study on the orchestration audio similarity per se.

Aim of the orchestration replication algorithm is to allow the creation of novel music pieces that share similar instrumental structure with another recorded piece. This similarity should be perceived on the level of sound texture of consecutive segments that constitute the piece. Information about tonal or rhythmic similarity should be discarded, since in the context of this work the automatic music composition system is supposed to be allowed to create compositions with no such constrains. The fact that the T_i, R_i and X_i compositions were composed with a similar algorithm, allows the examination of the orchestration replication of our system as stated in this paragraph, summing up to a single question: are the recordings in R_i perceived more similar to the ones in T_i than the recordings in X_i? To this end we need to utilize a *sound texture perceptual distance* denoted as $D_f(A, B)$ that captures the differences between two recordings, A and B according to a feature f.

For a formulation of the aforementioned sound texture perceptual distance, as mentioned previously, we have used the Mel-Frequency Cepstral Coefficients (MFCCs), the Bark scale frequency banks and the total loudness computed with Stevens' method. The features extracted with these tools are intended to capture not only the overall saturation in the respective MFCCs and Bark banks, but also the similarity of their fluctuations throughout the entire recordings. For their implementation we have used a part of the routines from the MA Toolbox for MATLAB [8], which provides several tools for audio music similarity.

We consider the matrix of the MFCC features of a piece P and denote it by $M_i^j(P)$, $i \in \{1, 2, \ldots, 12\}$, $j \in \{1, 2, \ldots, 3013\}$, where the index denotes the row and the exponent the column of the respective element. Similarly, we denote as $B_i^j(P)$, $i \in \{1, 2, \ldots, 20\}$, $j \in \{1, 2, \ldots, 3013\}$, the matrix of the Bark scale features of a piece P. Moreover, the mean value of the elements of each row of a matrix K_i^j is denoted as $\mu_i(K_i^j)$ and of the elements of each column by $\mu_j(K_i^j)$. Using the aforementioned denotations and by denoting the Euclidean distance as $|\cdot|$, we define the following distance measures:

1. **v–M:** Vertical MFCC means difference.
$$D_{\text{v–M}}(X, Y) = \left|\mu_i\left(M_i^j(X)\right) - \mu_i\left(M_i^j(Y)\right)\right|$$

2. **h–M:** Horizontal MFCC means difference.
$$D_{\text{h–M}}(X, Y) = \left|\mu_j\left(M_i^j(X)\right) - \mu_j\left(M_i^j(Y)\right)\right|$$

3. **c–M:** Horizontal mean of MFCC differences.
$$D_{\text{c–M}}(X, Y) = \left|\mu_j\left(M_i^j(X) - M_i^j(Y)\right)\right|$$

4. **v–B:** Vertical Bark means difference.
$$D_{\text{v–B}}(X, Y) = \left|\mu_i\left(B_i^j(X)\right) - \mu_i\left(B_i^j(Y)\right)\right|$$

5. **h–B:** Horizontal Bark means difference.
$$D_{\text{h–B}}(X, Y) = \left|\mu_j\left(B_i^j(X)\right) - \mu_j\left(B_i^j(Y)\right)\right|$$

6. **c–B:** Horizontal mean of Bark differences.
$$D_{\text{c–B}}(X, Y) = \left|\mu_j\left(B_i^j(X) - B_i^j(Y)\right)\right|$$

7. **Nt:** The difference of the total loudness of each time frame (as divided for the MFCCs and the Bark scale features) between two pieces using Stevens' method.
$$D_{\text{Nt}}(X, Y) = |L(X) - L(Y)|,$$
where each element of the $L(P)$ vector computes the power of the respective time window.

The aforementioned distance measures are computed for the respective pairs of pieces between the sets $D_f(T_i, R_i)$ and $D_f(T_i, X_i)$, with $i \in \{1, 2, \ldots, 30\}$. Table 2 demonstrates the mean value of these distance measures for the respective sets. The mean value of all distance measures between the T_i and R_i is smaller the one between the T_i and X_i sets. Even though the instrumentation in both cases (i.e. $D_f(T_i, R_i)$ and $D_f(T_i, X_i)$) is the same for all the pairs on which the distance is computed (i.e. for each i), the replication of the intensity variations effected the distances even on the "vertical" measures (i.e. v–M and v–B). As expected, the "horizontal" distance measures have yielded greater differences, especially the total loudness difference. These results indicate that the proposed approach allows us to produce ACSs that compose music and to some extent replicate the orchestration of a target recording.

7. CONCLUSIONS AND FUTURE ENHANCEMENTS

This work discussed the utilization of Horizontal Orchestration Replication (HOR) methods based on multichannel music recordings. HOR introduces the

Table 2: Mean distance for the considered features among the respective 90 recordings in the target (T_i), replicate (R_i) and random (X_i), $i \in \{1, 2, \ldots, 30\}$ sets.

f	$\mu(D_f(T_i, R_i))$	$\mu(D_f(T_i, X_i))$
v–M	20.08	21.35
h–M	350.30	373.00
c–M	56.55	59.00
v–B	3.36	5.23
h–B	77.15	99.37
c–B	9.53	11.74
Nt	284.00	371.70

utilization of information about the intensity variations of instruments that comprise a target recording to create novel algorithmic music with similar orchestration. This technique allows the underlying Automatic Algorithmic Composition System (AACS) to compose music independently, in contrast to other orchestration systems that act on the vertical domain. Moreover, an algorithm which computes the Intensity Variation Curves (IVCs) of single–track recordings of each instrument is described. This algorithm was tuned using DE on recordings of instruments with diverse timbres and intensity level plans.

The discussion additionally involved the description of an evaluation scheme for such orchestration methods, which is based on the similarity accuracy required for effective orchestration replication. The evaluation scheme introduced was applied for assessing the efficiency of the proposed horizontal orchestration scheme. Initial results indicate that the proposed intensity variation extraction method produces orchestrations that are perceived more similar than the ones produces by random intensity variations, but the lack of relative methods makes it difficult to conclude whether the proposed methodology produced optimum results.

The extraction of the IVC of an instrument through its waveform seems abstract and case dependent, since a recording strongly depends on the rhythmical patterns and accentuation intensity variations. Further improvements are needed for better evaluation accuracy of the IVC extraction algorithm described in Section 5. Having tunable parameters, the aforementioned algorithm could probably be specially adjusted to produce better results for clusters of instrumental timbres. For example, certain sets of parameters could be used for extracting proper IVCs for instruments with short attack times (piano, pizzicato, vibraphone), low or high spectrum content, etc.

The aforementioned algorithm can be also applied on larger data sets, with more diverse timbres. In parallel, the horizontal orchestration replication of a target recording can be attempted with the utilization of an instrument separation algorithm and the proposed IVC extraction algorithm. A system could thus be produced that automatically composes music and then orchestrates following the target input music content.

Additional future directions may include the extraction of information related not only to intensity level variations, but also to rhythmical density and melodic tension. Ultimately, a system that utilizes a library of orchestrations could be formulated. This system, using computational intelligence methods, would manipulate the IVCs of clusters of recordings with stylistic similarities and orchestrate novel musical works of certain styles.

8. REFERENCES

[1] Grégoire Carpentier and Jean Bresson. Interacting with symbol, sound, and feature spaces in orchidée, a computer-aided orchestration environment. *Comput. Music J.*, 34:10–27, March 2010.

[2] Grégoire Carpentier, Damien Tardieu, Jonathan Harvey, G?rard Assayag, and Emmanuel Saint-James. Predicting timbre features of instrument sound combinations: Application to automatic orchestration. *Journal of New Music Research*, 39(1):47, 2010.

[3] B. G. Churcher. Calculation of loudness levels for musical sounds. *The Journal of the Acoustical Society of America*, 34(10):1634–1642, 1962.

[4] Maximos A. Kaliakatsos-Papakostas. Personal web page.
https://sites.google.com/site/maximoskp/.

[5] Anssi Klapuri, Tuomas Virtanen, and Toni Heittola. Sound source separation in monaural

music signals using excitation-filter model and em algorithm. In *ICASSP'10*, pages 5510–5513, 2010.

[6] Paul Mermelstein. Distance Measures for Speech Recognition–Psychological and Instrumental. In *Joint Workshop on Pattern Recognition and Artificial Intelligence*, 1976.

[7] Eduardo Reck Miranda and John Al Biles. *Evolutionary Computer Music*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[8] E. Pampalk. A Matlab Toolbox to Compute Music Similarity from Audio. In *Proceedings of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[9] Dirk-Jan Povel. Melody generator: A device for algorithmic music construction. *Journal of Software Engineering and Applications*, 3(7):683–695, 2010.

[10] David Psenicka. SPORCH: an algorithm for orchestration based on spectral analyses of recorded sounds, 2003.

[11] François Rose and James E. Hetrick. Enhancing orchestration technique via spectrally based linear algebra methods. *Comput. Music J.*, 33:32–41, March 2009.

[12] R. Storn and K. Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[13] David Temperley. Melisma stochastic melody generator.
http://www.link.cs.cmu.edu/melody-generator/.

[14] David Temperley. *The Cognition of Basic Musical Structures*. The MIT Press, September 2004.

[15] B. A Trenholm. A least squares algorithm for fitting piecewise linear functions on fixed domains. Technical report, September 1985.

[16] George Tsihrintzis, Paraskevi S. Lampropoulou, and Aristomenis S. Lampropoulos. Musical instrument category discrimination using Wavelet-Based source separation. In *New Directions in Intelligent Interactive Multimedia*, volume 142 of *Studies in Computational Intelligence*, pages 127–136. Springer Berlin / Heidelberg, 2008.

[17] Iannis Xenakis. *Formalized Music: Thought and Mathematics in Composition (Harmonologia Series, No 6).* Pendragon Pr, 2nd edition, March 2001.

[18] E. Zwicker. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.