Generation of Two-Voice Imitative Counterpoint from Statistical Models

Victor Padilla[1]*, Darrell Conklin[2]

[1] UNIR, Universidad Internacional de La Rioja, Logroño (Spain)
[2] Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain and IKERBASQUE: Basque Foundation for Science, Bilbao (Spain)

Received 24 July 2018 | Accepted 18 September 2018 | Published 6 October 2018

Abstract

Generating new music based on rules of counterpoint has been deeply studied in music informatics. In this article, we try to go further, exploring a method for generating new music based on the style of Palestrina, based on combining statistical generation and pattern discovery. A template piece is used for pattern discovery, and the patterns are selected and organized according to a probabilistic distribution, using horizontal viewpoints to describe melodic properties of events. Once the template is covered with patterns, two-voice counterpoint in a florid style is generated into those patterns using a first-order Markov model. The template method solves the problem of coherence and imitation never addressed before in previous research in counterpoint music generation. For constructing the Markov model, vertical slices of pitch and rhythm are compiled over a large corpus of dyads from Palestrina masses. The template enforces different restrictions that filter the possible paths through the generation process. A double backtracking algorithm is implemented to handle cases where no solutions are found at some point within a generation path. Results are evaluated by both information content and listener evaluation, and the paper concludes with a proposed relationship between musical quality and information content. Part of this research has been presented at SMC 2016 in Hamburg, Germany.

Keywords

Artificial Intelligence, Music Informatics, Music Generation, Sequential Pattern Mining, Statistical Models Of Music.

DOI: 10.9781/ijimai.2018.10.002

I. Introduction

In music history, when a style ends and is replaced by a new way of thinking, theorists and musicologists try to explain the “obsolete” musical ideas. It is at this moment when the old style is classified, and the rules of the old style are systematized for future generations of musicians. According to Jeppesen [1], the history of music theory and style are far from being identical, and it is important to take into account the constantly recurring mistakes of the theorists with regard to the description of musical style. The principal inaccuracies of theorists according to Jeppesen [1] are:

1. An inclination that is common to these writers to theorize on their own account.
2. The moment of inertia which causes theorists to transfer rules from older textbooks to new without proper critical revision.
3. Inability of theorists, when describing the practices of past times, to discriminate between these and the elements of style typical of their own contemporaries, (which was the case with Fux).
4. Pedagogic considerations, which often tend to a simplification or relaxation of the set of rules belonging to the style, but often also to a stricter rendering of these rules “for the sake of exercise”.

Perhaps the most famous example of this theoretical approach is the “Gradus Ad Parnassum” of Johann Joseph Fux (1725). Fux presents a pedagogical method that breaks the learning task into well-defined graduated stages, from note against note through to florid counterpoint. This continues to be a standard pedagogic counterpoint text. However, for modern music informatics, generating music based on such stylistic rules is not a good approach: in fact, music from Renaissance to Romanticism can be written following basically the same rules. For example, stylistic differences between Bach and Palestrina counterpoint cannot be completely defined by basic generic rules (for example, prohibition of parallel fifths or octaves, voice leading at suspensions, etc.), and implementing specific exceptions to them can be a very complex task. The rule-based system achieved will become brittle and imprecise. Musical style must be learned from examples in order to model as closely as possible a musical style.

The corpus of pieces we are working with comprises 717 movements from Palestrina masses, comprising almost 350,000 vertical dyads (slices), providing a massive amount data for training statistical models of counterpoint. Even though these data are naturally within the musical style, composing a piece of music in the Renaissance style is not so simple as performing random walks through a statistical model. Counterpoint is full of imitations, canons, motifs and augmentations, and such devices cannot be captured by a first-order Markov model trained on limited data [2]. For solving these limitations and to provide coherence to the generated pieces, we take a piece from the corpus, referred to as a *template*, and discover its repeated patterns based on different viewpoints. The discovered patterns are used to cover the template piece and are adhered to in new music generated by the statistical model.

There are several arguments for using the masses of Palestrina as a test collection for our system. They are a model for a standard Renaissance style in counterpoint. Many universities and conservatories teach

* Corresponding author.

E-mail address: victor.padilla@unir.net

this style as basic training for new students in composition. Another important aspect is the homogeneity of the corpus of pieces. There are no significant differences in style between the first and the last mass, and the number of pieces is big enough to build a probabilistic model.

At the end of the article, we present some methods for the evaluation of the results. To assess an automatic system of musical composition is always a difficult task and in this paper we use two diverse alternatives: a listener evaluation in a live performance setting, and an objective information content method. For the listener evaluation, even when asking an expert group in a "Turing test" setting, the results cannot be very reliable, as they depend on many variables such as the performance, involvement of the experts, and their degree of knowledge of the musical style. An objective way to measure the relative quality of generated pieces is by using the information content of the generated piece, which is the negative log probability according to the statistical model.

This article is organized as follows: Section II reviews different works for generation and analysis of music. Section III describes the corpus of Palestrina masses. Section IV explores the concept of probability with respect to zero- and first-order Markov models of the Palestrina corpus. For obtaining patterns from a template, slicing the scores into dyads, and generating from a statistical model, we are using the concept of horizontal and vertical viewpoints developed and refined by Conklin [3]-[6]. Section V describes how to apply restrictions imposed by the template during music generation. Section VI presents and analyses the results obtained, and Section VII positions our work within previous research specifically on counterpoint analysis and generation.

II. Prior Work

Artificial Intelligence is a mature field having a broad variety of applications, ranging from driverless cars, natural language and speech processing, and computer players for board games, but is hard to define in formal terms for of Computational Creativity [7]. Often the scientist and the artist speak different languages and have different goals in mind, but this provides an interesting breeding ground for research with a lot of challenges to solve. Since the 1950s different techniques from artificial intelligence have been used for algorithmic composition and music generation. In this section we comment on some of the milestones of music generation, divided into two groups: rule-based approaches and machine learning approaches. In many cases, the lines between some methods are blurred and employ a mixture of both types of model creation.

A. Ruled-based

The generation of new music based on rules has a long tradition, from the works Hiller & Isaacson [8] using the ILLIAC computer at the University of Illinois. This early work was designed as a series of experiments on music composition, and rules for counterpoint were used in the generation of the first and second movements of the Illiac Suite. Rothgeb [9] encoded rules of eighteenth-century harmony for specifying adequate chords given bass notes. Many works on logic programming can be formulated as constraint satisfaction problems (CSPs). In this area, Ebcioglu [10] implemented rules for counterpoint translating constraints of fifth-species counterpoint to Boolean functions. In 1988 he subsequently developed 350 rules for the harmonic and melodic generation of Bach chorales [11]. Following Ebcioglu's work, several lines of research have been developed for harmony or counterpoint. Tsang & Aitken [12] harmonize four-part chorales, and Ovans & Davison [13] create a CSP system for the first-species counterpoint. Ramirez & Peralta [14] build a constraint logic programming system, for harmonizing a melody. Phon-Amnuaisuk [15] implemented a constraint system for harmonizing chorales in the style of J. S. Bach, adding control over the harmonization process in a flexible way. In recent years, Herremans & Sorensen [16] work with different counterpoint species using a variable neighborhood search algorithm. Komosinski & Szachewicz [17] address the difficulty of evaluating penalty (or reward) values for each broken (or satisfied) rule. The use of an additive function counting broken rules is known to have several drawbacks. First of all, it assumes that one can somehow determine the importance of each rule Another important drawback is that breaking one very important rule is equivalent to breaking several less important rules. Therefore, simple additive rule weighting function is found to be weak, and they propose to use a dominance relation. The implementation of rules by fuzzy logic to generate two-voice first species counterpoint is analysed by Yilmaz and Telatar [18].

Expanding the idea of "rule", a formal grammar may be viewed as a set of rules that expand high-level symbols into more detailed sequences of elements, in the same way that a language is constructed in a hierarchical structure of linguistic constituents. Some efforts to codify rules by hand and extend them to a grammar for automatic composition were made by Roads [19], Holtzman [20] and Jones [21]. In the 1990s some systems appear that hybridize rule-based systems with evolutionary algorithms using a fitness function from a rule set. McIntyre [22], Horner & Ayers [23] and Phon-Amnuaisuk et al. [24] worked in this direction in the context of four-part harmonization. For species counterpoint, Polito et al. [25] extracted rules and used them to define a fitness function followed by agents that cooperate to produce the composition. Gwee [26] worked with species counterpoint and a fitness function based on fuzzy rules.

B. Machine Learning

Machine learning is the subfield of computer science that studies the ability of computers to learn without being explicitly programmed. Many different machine learning techniques exist and for music generation learning is mainly unsupervised (without negative examples). In this section, we will survey some research on music generation using grammatical inference, Markov chains, and Artificial Neural Networks.

1) Grammatical Inference

As noted earlier, a grammar can be defined as a set of rules formally describing a language. The problem with a grammatical approach to composition is the difficulty in defining the rules manually. To tackle this issue, Cruz-Alcázar & Vidal-Ruiz [27] implemented several methods of grammatical inference inducing stochastic regular grammars to parse the compositions and make new pieces. Gilbert & Conklin [28] present a method to find tree structures in musical scores using a *probabilistic context-free grammar* for melodic reduction. The method is applied to parse phrases from Bach chorale melodies, and the statistical model is also used to evaluate the information content of the pieces. Following the ideas of Gilbert & Conklin, Groves [29] explores the generation of melodies from a probabilistic analytical model of melodies. Quick & Hudak [30] present a new class of generative grammars called *probabilistic temporal graph grammars* to handle temporal aspects of music in a way that retains a coherent metrical structure.

2) Markov Chains

Markov Markov chains are stochastic processes transitioning in discrete time steps through a finite set of states. In music composition, the transition matrices may be trained from a corpus of pre-existing compositions. In an order-n Markov chain, the next state depends on the last n states. In a hidden Markov model (HMM), the states are hidden and the goal is to infer an optimal state sequence for an observed sequence. This approach has been studied by Farbood &

Schoner [31] who train a second-order HMM to generate Palestrina-style first-species counterpoint to a specified cantus firmus line. The method uses Markov chains which capture the rules of counterpoint using probabilistic tables for harmony, melody, parallel motion, and cadences. Herremans et al. [31] use a first-order Markov model from a corpus of first species counterpoint and compare the ability of variable neighborhood search, iterative random walk and Gibbs sampling, to generate a hidden counterpoint line. Results are evaluated by information content (average negative log probability of the fragment using the dyad transitions of the transition matrix).

Working with HMMs and Bach chorales, Allan & Williams [33] create a system to compose four voice textures given a soprano part. The chord sequence is generated using the Viterbi algorithm, and for the passing notes, a second HMM is employed. Whorley & Conklin [34] present a multiple viewpoint system of four-part harmony to evaluate and improve an iterative random walk technique. This is evaluated using information content and also with a small set of rules of harmony.

3) Artificial Neural Networks

Artificial Neural Networks (ANNs) are biologically inspired models made of interconnected sets of nodes in several layers. Some of them are the input connections while others have output signals with several interconnected layers between.

Focusing on polyphony and counterpoint, Hild et al. [35] develop a model designed to solve a four-part chorale harmonization in Bach's style. The system was called HARMONET and had a three-layered architecture. An evolution of HARMONET was MELONET [36] and improved by Hörnel & Degenhardt [37]. Since then, many hybrid systems have been developed, for example NETNEG [38] which used an ANN trained with sixteenth-century classical music compositions. That generates melodic segments and polyphony was generated by a rule-based system of agents. Verbeurgt et al. [39] join Markov chains for constructing motifs and a trained ANN to assign the absolute pitches. Adiloglu & Alpaslan [40] used the back-propagation algorithm to generate two-voice first species counterpoint pieces. Some researchers have recently applied methods from deep learning to the chorale harmonization problem [41][42].

C. Generation Using Patterns

As suggested by Conklin [2], the construction of computational methods for musical style imitation has been far more difficult than initially imagined. Listening to music and perceiving its structure is an easy task for specially trained musicians but building computational models to mimic these processes is a hard problem. The use of patterns during generation can help to ensure coherence and intra-opus repetition in generated pieces. A method for the detection of melodic phrases in the masses of Palestrina is described by Knopke & Jurgensen [43], based on the use of suffix arrays to find repeated patterns. Sidorov et al.[44] present an approach to music analysis, in which an inferred grammar is explains the structure of a musical work.

Data mining is a process of extracting small pieces of valuable information from large data. A special situation is when data is in the form of sequences, and several sequential pattern mining methods have been developed in the last decade [45]. Using the idea of viewpoints [2]-[5], music can be converted into a string (or parallel strings) of features and analysed using sequential pattern mining methods. In our research, for analysing patterns, we are using the gap-BIDE algorithm [46] with zero gaps between sequences. This will be further explained in Section IV.B.

Some researchers have developed methods for combining patterns and constraints with Markov models. Pachet, Roy and Barbieri [47] try to solve unary and adjacent binary constraints with Markov models using arc-consistency techniques and re-normalization. Collins et al. [48] describe and evaluate a computational model of stylistic composition using discovered patterns to constrain a Markov model of vertical slices. Conklin [5], focused on trance music, explores a new approach to generating high probability and coherent chord loops from a statistical model trained on a chord sequence corpus. David Cope's Experiments in Musical Intelligence [49] is a system for algorithmic composition heavily based on the conservation of patterns.

III. Palestrina's Masses

The style of Palestrina can be seen as a combination of melodic lines in a polyphonic environment, characterized by the tension between harmonic and melodic elements. The line is the starting point of Palestrina's style, and the harmony does not have an independent sphere of interest as in Bach counterpoint. Jeppesen [1] analyses Palestrina's style using different aspects such as rhythm, modes, lyrics, melody, harmony and dissonance. His point of view is very clear about the secondary role played by harmony: "…The exactions arising out of harmonic aspects are really only intended to ensure the sonority of the individual harmonic moments. ´Harmonic´ does not signify here any independent sphere of interest; chords had not yet reached a stage when they had their own vigorous life, as in Bach's works". All that is required in vertical chords is clearness and sonority. The imitation is the base for constructing the polyphony. Accidentals are limited to F, G and C-sharp and B and E flat, found in plainsongs and related to Gregorian modes. De la Motte [51] mention that, "Palestrina just polished and refined a language developed by Josquin 70 years before". The accidentals of Palestrina are a natural evolution of the polyphony of Josquin des Prés, where E flat was always associated with B-flat and C, F and G sharp embellishments of the notes D, G and A.

The corpus of pieces we are working with consists of 101 masses composed by Palestrina (see Table I). These masses were published between 1554 and 1601, after his death in 1594. The date of composition of the different pieces is very difficult to determine, and each mass consists of various movements: Kyrie, Gloria, Credo, Sanctus, Benedictus, Agnus Dei. Each movement is divided into sections based on the text. The masses and the movements vary in the number of voices from three to six. For example, Benedictus in many masses is written in three voices and Kyrie in five or six. Table I describes is the corpus of pieces we have, using the data of music21 [52], a Python-based toolkit for computer-aided musicology developed by MIT.

Taking into account just two voices, the number of vertical slices available is almost 350,000 which provides enough information for constructing a reasonably accurate first-order Markov model, as is explained in the next section.

Table I. Corpus of Pieces of Palestrina from music21

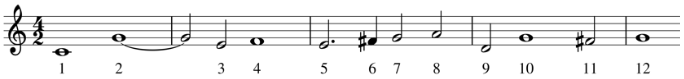
Mass part	Pieces
Agnus	186
Benedictus	99
Credo	98
Gloria	101
Kyrie	129
Sanctus	104
Total:	**717**

IV. Viewpoints for Pattern Discovery

For the generation of polyphony, both horizontal (melodic) and vertical (harmonic) aspects must be modelled and we use the concept of *viewpoints*, developed and refined by Conklin [2]-[5], from the horizontal and vertical perspective. In our work, the generation of imitative counterpoint is developed at two levels, corresponding to the short- and long-term models of viewpoints. Long term or stylistic aspects are modelled using vertical slices and short term or intra-opus aspects using patterns discovered in a template piece. A template piece of music is transformed into a higher level description derived from the basic surface representation, by converting the sequence of basic events into sequences of derived viewpoint elements. A linked viewpoint is a combination of two or more viewpoints that models their interaction simultaneously. Following these steps, pattern discovery is performed on the transformed representation.

A. Horizontal Viewpoints

Each voice of the chosen template piece is cut into phrases which are assumed to be separated by rests. This division of phrases is possible in Palestrina vocal music, where music is thought taking into account the phrases of the text, and rests are written between separate ideas, never as part of a musical idea. Once the score is divided, each phrase of Palestrina music is treated as a sequence of linked viewpoint values. To better understand the concept of viewpoint, we take a melody of Palestrina. The sequence of notes is converted to a sequence of features derived from the musical surface (Fig. 1), for example, absolute pitch (pitch), name of note (spell), melodic contour, duration contour, interval (diaintc), or an abstract interval class (scalestep), as will be explained below. A pattern is a sequence of features $(v_{1,\dots},v_i)$ where each v_i is a feature (e.g. scale step linked with duration contour).



	1	2	3	4	5	6	7	8	9	10	11	12
pitch	60	67	64	65	64	66	67	69	62	67	66	67
spell	C	G	E	F	E	F#	G	A	D	G	F#	G
contour (pitch)		+	+	+	-	+	+	+	-	+	-	+
contour (dur)		+	-	+	-	-	+	=	=	+	-	+
diaintc		J5	m3	m2	m2	M2	m2	M2	J5	J4	m2	m2
scalestep		J45	Mm3	Mm2	Mm2	Mm2	Mm2	Mm2	J45	J45	Mm2	Mm2

Fig. 1. Different viewpoints applied to a melody of Palestrina. Agnus from Beata Marie Virginis. altus, bars 5 to 9.

The scalestep viewpoint groups successive intervals and is flexible enough to find patterns in Renaissance style. The values of that viewpoint are:

- Unison and Octave (J18)
- Minor second and Major second (Mm2)
- Minor third and Major third (Mm3)
- Perfect fourth and Perfect fifth (J45)
- Minor sixth – Major sixth (Mm6)
- Minor seventh - Major seventh (Mm7)

The repetitions of patterns in Palestrina are not merely exact transpositions of intervals. For example, a minor second can be converted to a major second, as is shown in Fig. 2. Using the syntax above, and taking into account just the scalestep viewpoint, the pattern indicated in Fig. 2 would be represented as:

$$\text{J45, Mm3, Mm2, Mm2, Mm2, Mm2, Mm2} \tag{1}$$

This pattern has seven components and represents the boxed segment in Fig. 2 (the soprano, altus and bassus of the Agnus from Beata Marie Virginis).

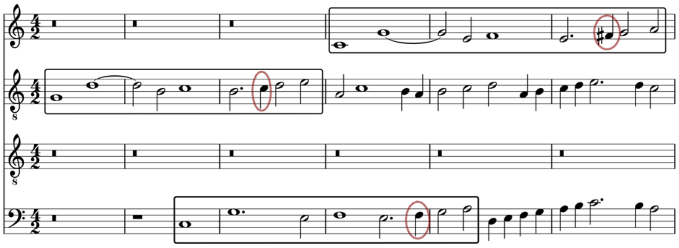


Fig. 2. Agnus from Beata Marie Virginis, bars 1 to 6. Palestrina. Pattern with different intervals.

B. Pattern Discovery

Data mining is the computational process of discovering interesting patterns in large data sets. This interdisciplinary subfield of computer science is growing, and the number of algorithms and researchers in the field highlights its importance. Sequential pattern mining has become an essential data mining task, with broad applications, including market and customer analysis, web log analysis or pattern discovery in protein sequences A survey on sequential pattern mining and the approach of the different algorithms, addressing efficiency and scalability, has been summed up by Khan & Jain [43].

Algorithms for sequential pattern mining include SPADE, Sequential PAttern Discovery using Equivalence classes [50], PrefixSpan, Prefix-projected Sequential pattern mining [51] , GSP, Generalized Sequential Pattern algorithm [52] CloSpan, Closed Sequential pattern mining [53], BIDE, BI-Directional Extension [54] or SPAM, Sequential Pattern Mining using A Bitmap Representation [55]. In our experiments we are using gap-BIDE [44], an extension of the BIDE algorithm for mining closed sequential patterns with possible gap constraints. Currently, we are working at zero gap level without taking into account gaps in the sequences.

To apply sequential pattern mining to Palestrina masses, each piece in the corpus is converted to a viewpoint sequence, with phrase boundaries indicated by rests, as explained in Section IV.A. The linked viewpoint for discovering patterns is:

$$\text{scalestep} \otimes \text{contour(dur)} \tag{2}$$

This particular linking of viewpoints allows the discovery of flexible patterns, e.g. augmented and diminished patterns. The viewpoint can also capture inversions, though it should be noted that equivalence according to the viewpoint is a necessary but not a sufficient condition for inversion. The vocal lines in Palestrina are very flexible and the imitations are sometimes just hinted at. Fig. 3 contains examples of patterns found in the Agnus II from the mass

The pattern 91 is an inverted scale made by six short equal notes plus a long note. Pattern 3 is more complex, typical of Palestrina music, where neither the durations are exact nor the melodic contour, but an attentive listener can easily identify the similarity of the two instances of this pattern. Regarding duration and melodic contour, pattern 1 is clearest but possibly a human musicologist would have taken the previous note of this pattern dismissing the different duration and interval of the first note in the different presentations.

C. Ranking Patterns

A huge number of patterns can be typically found in a template piece: some trivial, and some method for ranking them is necessary [6]. In this paper we establish a ranking of patterns based on a binomial distribution that computes the probability of obtaining an observed number of occurrences in a given number of sequence positions within the template piece.

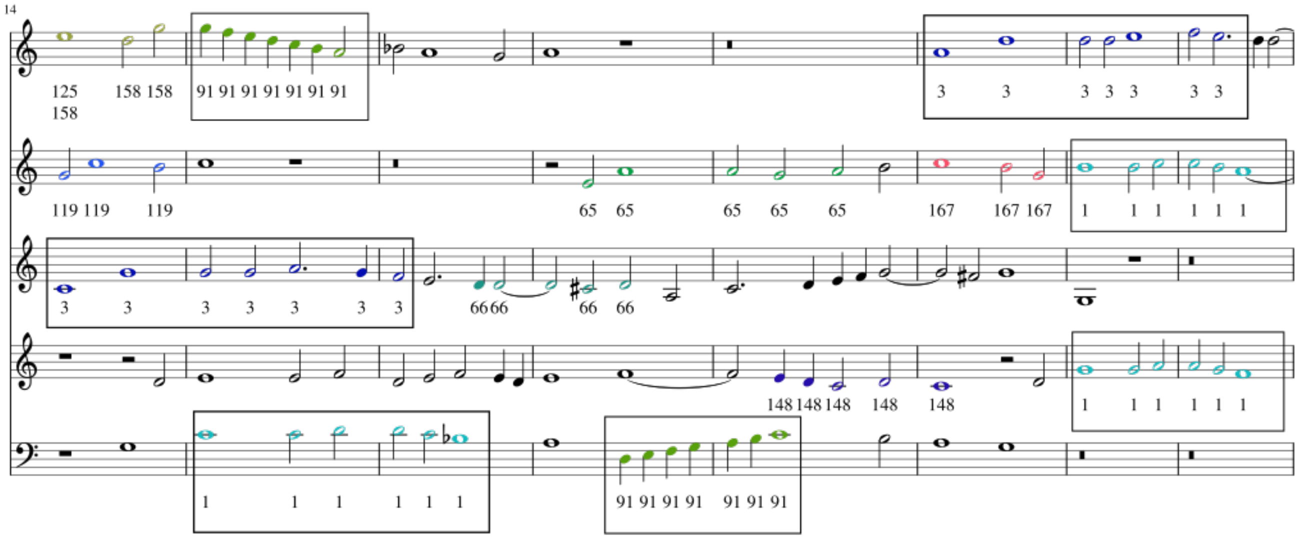
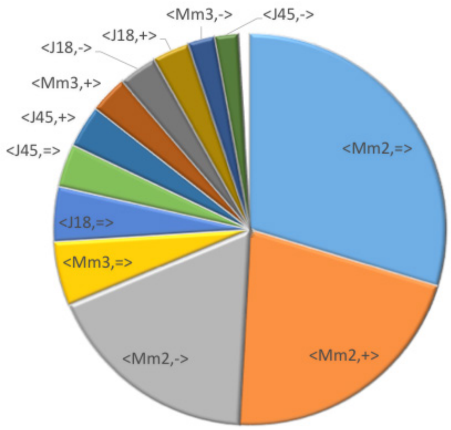


Fig. 3. Agnus II from Ascendo ad Patrem, bars 14 to 21. Palestrina. Patterns detail of patterns using scalestep ⊗ contour(dur).



Patterns
<J45,+>, <Mm3,->, <Mm2,+>, <Mm2,->, <Mm2,->, <Mm2,+>, <Mm2,=>
<J45,+>, <Mm3,->, <Mm2,+>
<J45,+>, <Mm3,->, <Mm2,+>, <Mm2,+>, <Mm2,->, <Mm2,+>

Fig. 4. A list of features and their counts (upper pie chart). Example of pattern encoding (lower table) from Beata Marie Virginis of Palestrina (see Fig. 1).

To rank patterns, it is necessary to know the background distribution of pattern components. Fig. 4 shows a distribution of the values of the viewpoint defined in (1). For example, <Mm2,=> indicates a scale step of a minor or major second with an equal duration, or <J45,+> indicates a scale step of perfect fourth or fifth where the second note has a higher duration. Clearly, the most probable interval is the second (69.4%) divided into same duration as the previous note (30.0%), higher duration than the previous note (21.4%) and lower duration than the previous note (18.0%). The rest of the melodic intervals have a much lower probability. This illustrates that, for example, patterns comprising predominantly <Mm2,=> features are not surprising and will not be significant unless they occur very frequently in the piece. The binomial pattern ranking, as described below, handles these effects in the piece.

The background probability of a pattern gives the probability of finding it in a random segment with the same zero-order distribution as the corpus. The background probability (of a pattern) using a zero-order model of the corpus is:

$$b_p = \prod_{(i=1)}^{l} c\,(v_i)/N \qquad (3)$$

where:

- $c(v_i)$ is the total count of feature v_i,
- N is the total number of places in the corpus where the viewpoint is defined.

Using the background probability of a pattern, its interest I can be defined using the binomial distribution which gives the probability of finding exactly k occurrences of the pattern in a sequence of length t, where the background probability is b. Then the negative log probability of finding at least the observed number of occurrences of the pattern.

$$I(p) = -\ln B(k;t;b) \qquad (4)$$

where:

- l_i gives the cumulative probability (right tail) of the binomial distribution,
- t approximates the maximum number of positions that can be possibly matched by the pattern,
- k is the number of times the pattern appears in the template piece.

with t calculated as follows:

$$t = \sum_{i=1}^{p} l_i - c + 1 \qquad (5)$$

where:

- p is the number of phrases,
- l_i is the length phrase i,
- c is the length of the pattern.

D. Building the Template

Fig. 5 is one example of different patterns found in one fragment of Agnus from Beata Marie Virginis of Palestrina, ordered by their interest value Eq. (3). The number followed by a colon (:) indicates the interest for each pattern. The same melody can be covered by patterns in many different ways. To cover a template, the ranking according to their interest value is used. If a lower pattern in the ranking overlaps with one higher, the lower pattern is not considered. This simple greedy algorithm is repeated until reach the last pattern, trying to complete the template. In this way, the template will be covered by non-overlapping interesting patterns. The template obtained will be used for creating the new piece as is explained in the next section.

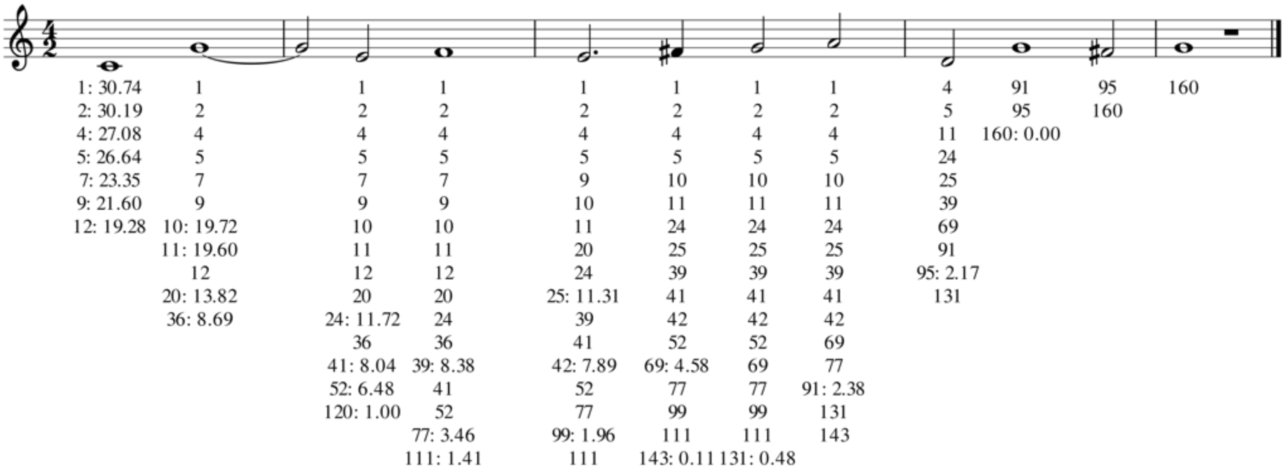


Fig. 5. Agnus from Beata Marie Virginis,of Palestrina. Altus, bars 5 to 9. Different patterns ordered by their interest (Equation 4). The first number gives the rank of the pattern, and the second the value for the first position of a pattern.

E. Vertical Viewpoints. Markov Model

For constructing the Markov model, two voices are selected and cut into slices (see Fig. 6). In this first approach, we have taken the highest and lowest voice for a better result, removing the inner voice. Usually, the music that follows harmonic constraints entrusts to the lower part (bass) an important role in the harmonic context, while the higher part (soprano) is more appropriate for defining melodies.

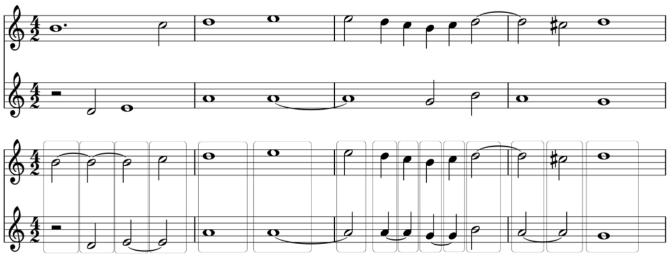


Fig. 6. An example of two-voice slicing.

The slicing process is the same as the full expansion method explained by Conklin [4], dividing when a new event appears in one voice. In our method, we do not retain ties between notes. In Renaissance vocal music, whether a note is repeated or tied sometimes depends on the text and furthermore durations are conserved from the template.

Taking into account pitch and duration, the number of slices in the corpus is 347,748. The zero-order Markov model is calculated counting the number of repeated slices and dividing by the total. The number of different slices is 1582 distributed as is shown in Fig. 7.

The vertical axis is the number of repetitions (logarithmic scale) and the horizontal the rank in the slice ordered by repetitions. Counting the number of next unique slices (first-order Markov model), also ordered by the number of repetitions (zero-order model), the results are shown in Fig. 8, where the number of different paths ranges from 0 and 183.

The piece now can be treated as a sequence of regular simultaneities where it is possible to apply different constraints that filter the possible paths. For example, based on the melody of Palestrina at the top of the Fig. 9, we illustrate the system with different restriction levels for creating a new upper voice. The upper voice is generated applying a random walk among the possible vertical slices using a first-order model. It is a short phrase, and it was easy to find solutions through forward generation with just one template and different viewpoint constraints in the lower voice. Ranking from strongest to weakest, and using linked viewpoints, they are labelled as pitch ⊗ duration, scalestep ⊗ duration and duration.

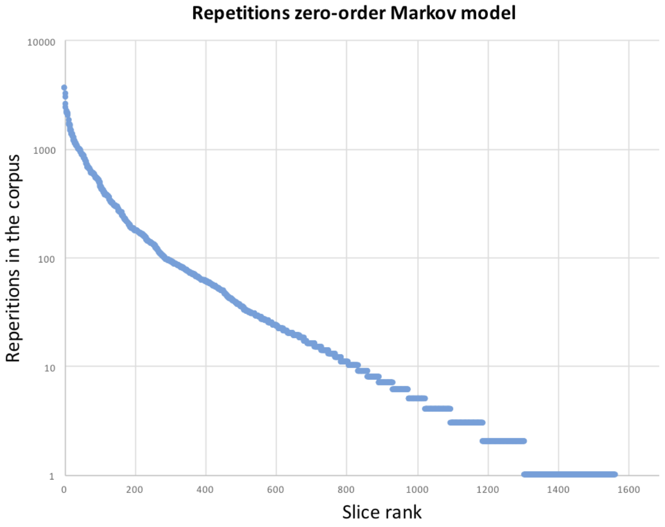


Fig. 7. Zero order distribution of repetitions.

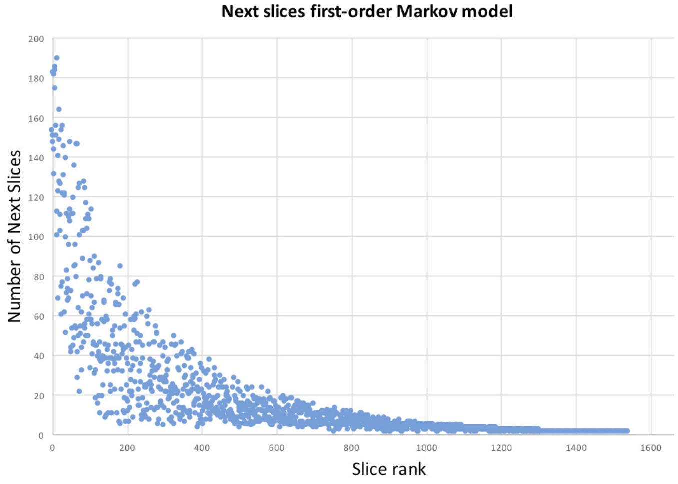


Fig. 8. Distribution of unique next slices, first-order Markov model.

V. Applying the Model to the Template

This section explains a method for generating new music based on a template which has been annotated with patterns, as described

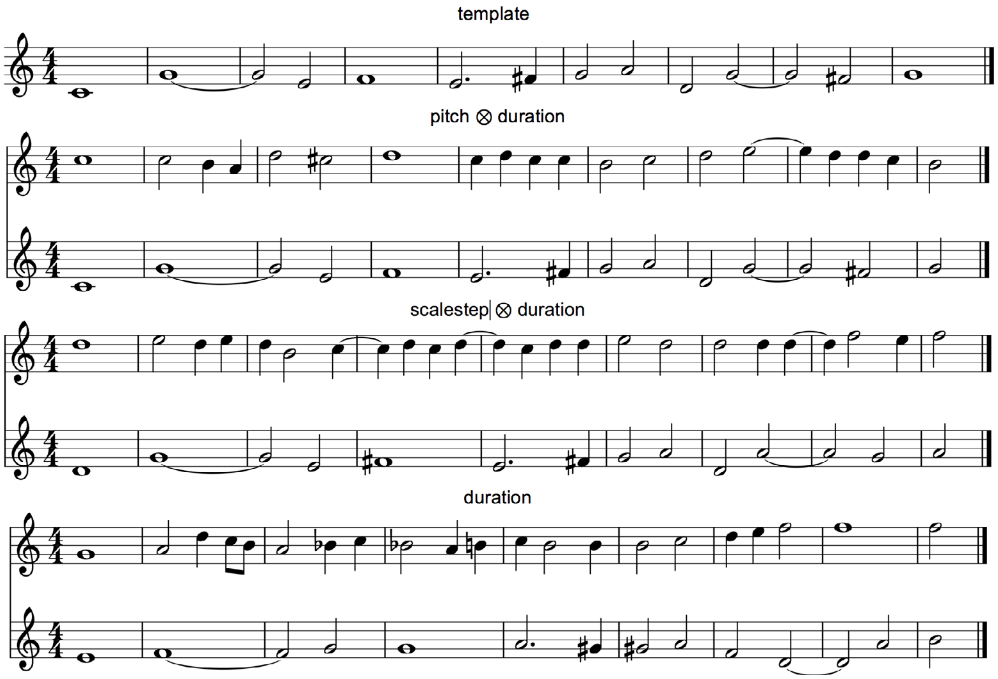


Fig. 9. Generation of upper voice based on different constraints in lower voice.

in Section IV. The idea is to fill the template with the slices and probabilistic paths obtained by the first-order model. The steps are as described below.

A. Forward Generation

For generating new music, one piece from the corpus is chosen, and patterns are discovered in the piece using the viewpoint scalestep ⊗ contour(duration) as mentioned earlier. Once the final template is constructed, for constraining notes within areas covered by patterns, the viewpoint

$$\text{scalestep} \otimes \text{contour(dur)} \otimes \text{contour(pitch)} \qquad (6)$$

is used. Note that this represents a slightly more restrictive linked viewpoint than that used for pattern discovery (1), in that the regions are also required to conserve pitch contour. The generated music, therefore, conserves the abstract qualities of scale step, duration contour, and pitch contour. Further, in this article, the exact rhythm from the Palestrina template is used, therefore, the conservation of duration is assured. For describing the method, we take Benedictus from the mass Descendit Angelus Domini as a template and proceed with the next steps:

- Remove internal voices retaining the highest and lowest.
- Divide the template into regions organized by the patterns. If the region is a pattern, the viewpoint shown in Eq. (6) is used for horizontal restrictions. If the region is not a pattern, just the duration viewpoint remains;
- Filter the vertical slices by the different constraints. If at one point it is not possible to find a next slice, a backtracking algorithm is performed (see Section V. B).

There is a probability associated with each piece, using the statistical model. Different results will be obtained choosing pieces with different overall probabilities, as will be commented in Section VI.

B. Backtracking Algorithm

Due to the severe restrictions forced by the template, it is possible to encounter some points where all slices to continue the piece have zero probability at the slices generated. This problem was due to the bottleneck arising from the availability of very few continuations for some slices of the corpus. To solve this problem, a double backtracking algorithm has been implemented at two different levels, pattern and template. At the pattern level, the system goes one, or several steps back if no possible solutions are obtained for some slice. If the backtracking at pattern level reaches the first slice, the system goes back one (or several steps back) from the patterns of the template. This method is faster and permits a scattered group of solutions uniformly distributed.

VI. Results and Evaluation

The method described in Section V have been used to generate new pieces based on the Benedictus from the mass *Descendit Angelus Domini* as the template. This Benedictus is composed of just three voices (as most of the Benedictus in Palestrina masses are). The main purpose of taking a three voices piece is that we have to remove just one staff, and the counterpoint, imitations, and harmony are less affected than in a four or five voices piece.

The evaluation of a system for generating music is always a complex task. If the generation is limited to a very narrow and particular type of composition (first-species counterpoint, i.e.), the evaluation can be defined regarding "how many rules have been broken". Some examples in this sense can be analysed in [34][18][40][31]. In the case of Palestrina, the evaluation of broken rules such as parallel fifths or octaves of two consecutive slices is not possible because slice transitions are taken directly from the corpus. It is therefore not possible to find parallel fifths or octaves of two consecutive slices unless they specifically are in the Palestrina corpus. Some unusual melodic movements in Palestrina style possibly appear, mainly related to the use of accidentals, but is very difficult to measure violations of accidental use automatically.

A. Computational Evaluation

Some research has found a close relation between *information content* and the "quality" of the results in a statistical model [3][34] The information content can be defined for a sequence of slices $e_{1,...}, e_n$ as follows:

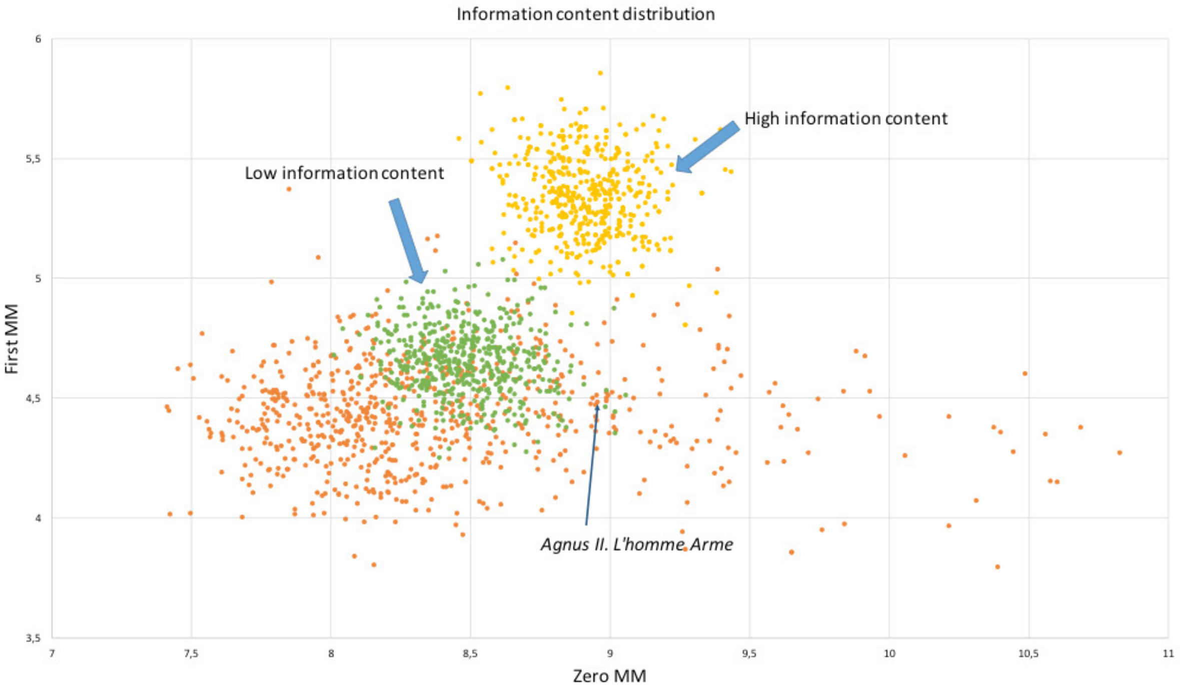


Fig. 10. Information content distribution using zero and first-order Markov Model. In orange the original corpus of pieces of Palestrina. In green and yellow the pieces generated based on the Agnus II from the mass L'homme Armé with low and high information content.

$$-\frac{1}{n}\sum_{i=2}^{n} \quad \log_2 P(e_i \mid e_{i-1}) \tag{7}$$

$P(e_i|e_{i-1})$ where is the probability of eventi e_i in the first-order Markov model.

Taking *Agnus II* from the mass *L'homme Armé* as a template, we have generated 1000 pieces distributed as follows:

- 500 pieces, choosing the next slice of the first-order Markov model from the upper third of the next slice distribution. This option will produce a group of solutions of low information content.
- 500 pieces, choosing the next slice of the first-order Markov model from the lower half of the next slice distribution. This option will produce a group of solutions of high information content.

The distribution of the original corpus of pieces and the pieces generated using the zero- and first-order Markov model is shown in orange in Fig. 10. In this figure, the pieces generated, based on the *Agnus II* from the mass *L'homme Armé*, are in green and yellow. The green cluster indicates the pieces generated with the low information content and the yellow one the pieces with high information content. The information content of the original template is 4.50 as indicated in Fig. 10.

Taking just the values of the information content of the first-order Markov model from the original template, the distribution is shown in Fig. 10, where the 1000 pieces generated are divided into two groups of entropy based on the probability of next slices. Choosing the next slice from the upper third, ranking them from highest to lowest probability (blue bars, left part), they are centred around 4.65, that is slightly higher than the information content of the original template, 4.50. Taking next slices from the second half of the ranking, the information content is clearly higher, centred around 5.4. This second group explores less probable links between slices.

B. Listener Evaluation

For listener evaluation, we have taken three pieces: the original Palestrina and one of each group of Fig. 11 (low information content, high information content). The information content of these pieces is 4.65 and 5.54 respectively. They were interpreted by a professional choir at "The Vortex Jazz Club" in London, which is a small concert hall with a maximum capacity of 100 people. The audience was asked to identify which of the three pieces was the original and asked for their degree of confidence in the decision, on a five-point scale. 55 questionnaires were returned. The results showed that 49.1% identified correctly the template and 50.9% were deceived, as is seen in Fig. 12. On the questionnaire were some a questions about musical knowledge like "*Do you play an instrument?, How long have you played or sung?, What types of music do you listen?*". Analysing the responses, 33/55 (60%) have played an instrument or sung for more than five years, and 40/55 (72.7%) listen to classical music, which indicates at least an average musical knowledge and a trained ear.

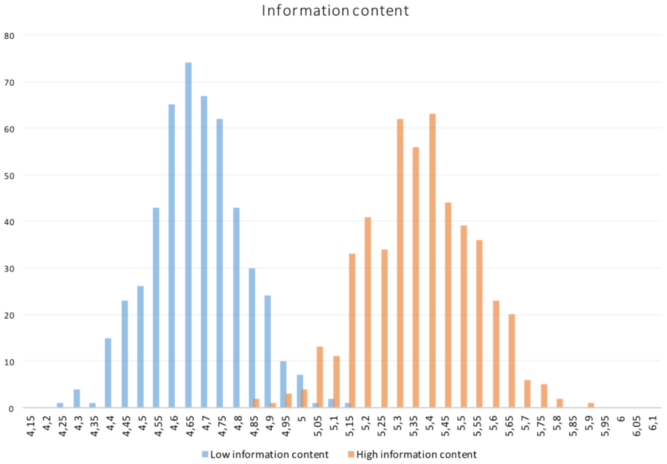


Fig. 11. Distribution of information content first-order Markov model of 1000 pieces generated divided into two groups of entropy based on the probability of next slices.

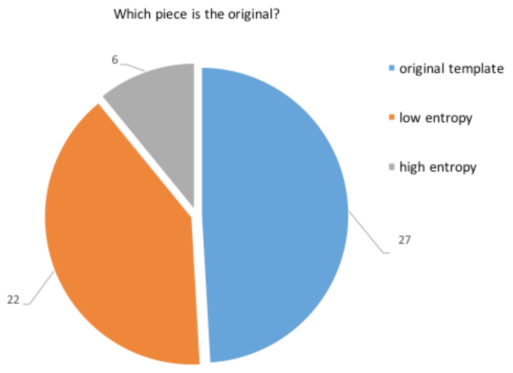


Fig. 12. Results of the question: which piece is the original?

The results obtained indicates a clear difference between the pieces generated with high and low entropy. Low entropy pieces are closer to the original style and even can deceive around the 50% of an audience with musical knowledge. It is possible that the melody and jumps in the vocal line of the high entropy piece are identified as "out of style" even without breaking counterpoint rules. Fig. 13 shows the results of the question, "How confident are you on a scale 1 to 5?".

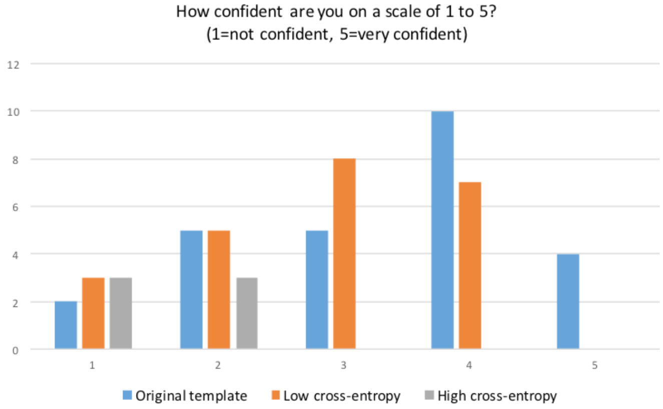


Fig. 13. Results of the question: How confident are you on a scale 1 to 5?

VII. Discussion

In this section, we take a closer look at the main milestones and the different approach to the counterpoint generation for comparing with our research. Starting from the past century, Gjerdingen [59] creates a computer program for counterpoint species writing rules by hand. Gjerdingen recognises that the counterpoint rules remind an Old Testament patriarch calling out the commandments, "…*Thou shalt have no tritones; Thou shalt not leap dissonance; Thou shalt not commit parallel fifths; and so on.*" Those rules are implemented using functions that control the melodic line and vertical intervals. The evaluation is made taking a *cantus firmus* and comparing examples of the book of Jeppesen & Glen [60]. Their main goal is to create small fragments of counterpoint, given a *cantus firmus*, following rules as closer as the examples of the book, but not to the real Renaissance style.

One step forward can be considered the work of Farbood & Schoner [31]. In this case, the rules are not implemented by hand. Each rule is implemented as a probability table where illegal transitions are described by probability zero. The transition probabilities for generating a counterpoint line are obtained by multiplying the individual values from each table, assuming the rules are independent. A database of species counterpoint (12 pieces composed by human and 44 generated by the computer) is used as training data for the Markov model. The research is limited to first species counterpoint, and the main objective is to infer rules from a corpus of pieces, not to generate florid imitative counterpoint.

Adiloglu & Alpasl [40] also work on first species, but in their case using neural networks combined with back-tracking algorithm. The input layer represents the notes coming from the cantus firmus and the output layer the new counterpoint generated. The result is evaluated in two steps, by the counterpoint rules (parallel fifths, and octaves) and by asking a group of musical experts. In this case, the rules are not inferred from the corpus as in the Farbood & Schoner [31] work. Adiloglu & Alpaslan [40] ask a group of experts about the quality of the music generated, a musicologist, a composer and a choir director. According to their opinion, the melodies generated are generally correct but there are some cases in which the rules were not obeyed. The music experts also commented that the melodies produced do not always sound interesting. It is important to underline that first species, note against note, is the very first exercise for the student in counterpoint. In our case, we are comparing real masterpieces against pieces automatically generated.

Herremans & Sörensen [16] develop a variable neighbourhood search (VNS) algorithm that can generate musical fragments of arbitrary length consisting of a first species counterpoint melody given a cantus firmus. The VNS is a local search algorithm that starts from a randomly generated melody and improves it by changing one or two notes at a time. When no improving fragments can be found in any of the neighbourhoods, a local optimum is reached. In order to get out of this local optimum, a perturbation strategy is used. The algorithm reverts back to the best found fragment and changes a predefined percentage of the notes to a random allowed pitch.

The previous research work with species counterpoint (mainly first species), and it is not taken into account the complexity of imitations and structure of the counterpoint. Knopke & Jürgersen [43] try to identify common melodic phrases in the masses of Palestrina mapping multi-character music symbols into single-character tokens to build a suffix array structure. This research is focused just on analyses and not in generation. They claim that this system identifies all transpositions, inversions, retrogrades and retrograde inversions of unknown melodic segments. One limitation is that they do not use abstract viewpoints, just pitch and rhythm, and slight changes in imitations are disregarded.

VIII. Conclusions and Future Work

This paper presents a method for generating new music based on the corpus of masses of Palestrina. To sum up, comparing our research with previous work, our research is made working with real pieces of Palestrina, regarding the complexity of the counterpoint. The species are simple exercises invented for training, but they are not proper compositions. An exercise in species is a short piece of music without any kind of coherence or imitation, just one *cantus firmus* (usually in whole notes) and a second melody in the same duration (first species) or shorter (rest of species). For the template, we use different combinations of linked viewpoints, much more flexible than simple pitch and rhythm. Small mutations in imitations are very common, and the patterns detected should be robust to these changes. Regarding the work of Knopke & Jürgersen [43], for the identification is required the exact matching of the patterns. For example, non-exact intervals (a fourth by a fifth or a third minor by major) or mutations in a melody, very common in counterpoint music, are not considered. Filling a template identifying imitation is a way of retaining some grade of coherence into the piece. The counterpoint is based on the motifs that are repeated in different voices. One possibility explored in this article is to have a template to fill. This reinforce a strong, but possible constraint that limits the number of possible paths. The employment of templates extracted from a piece is just one possibility for working with coherence, but this concept is nor even considered in the previous works.

The method, for practical purposes, is limited to two voices taking a template from the corpus without overlapping patterns. The gap-BIDE algorithm and the binomial distribution explained in Section IV.B works correctly detecting patterns. From an intuitive point of view, the ranking of the patterns discovered, in most of the cases, is related to the importance of the pattern in the piece. The greedy covering algorithm is quite simple and will be revised in a future version. Though this aspect is not the main goal of the project, a deeper research finding strengths and weaknesses of the method for finding patterns, template extracted and the covering algorithm should be done.

Regarding the Markov model, a first-order imodel s a good approach for ensuring correctly linked slices with rhythm and pitch constraints, preventing "weak" successive chords having grammatical errors such as parallel fifths, and parallel octaves, without implementing these devices using specific rules. This model does not organize harmonic regions, and "non-idiomatic" melodic movements can appear, mainly associated with accidentals. In this sense, a second-order model implementation could be an improvement for generating better melodies, but the training data would decrease exponentially. The main goal of this work is that the template complements some weaker aspects of the first-order Markov model and provides some kind of melodic coherence. In other systems, for example, David Cope's EMI [49], the coherence is achieved analysing bigger slices of the pieces, somehow inspired by the idea of *Musikalisches Würfelspiel* of some classical composers. In our case, the slices are reduced to the minimum rhythmic value and the possible structural information obtained, sparse. The template, therefore, provides the necessary scaffolding for the melodic ideas.

Section V.B commented on the double backtracking algorithm performed if no solution is found. The processing time is very high to find solutions using random walks when the group of optimum linked slices is very small, and in some cases, there may not be a solution due to the hard requirements of the patterns selected. The backtracking algorithm is faster than a simple random walk and provides a group of solutions homogeneously distributed. Another possibility that could be implemented in a future version is a depth-first search to explore all the different paths, which might lead to more heterogeneity in the results.

This model is made and tested for two voices due to the sufficient population of dyads in the corpus, but it is possible to extend to three or more voices using different viewpoints such as vertical intervals and duration. The zero-order Markov model will grow significantly, and the slices with higher probabilities will possibly decrease, augmenting the dead-end solutions, but hopefully, the corpus is large enough to find paths and create new and interesting pieces.

Acknowledgment

This research was supported by the project Lrn2Cre8 which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

References

[1] K. Jeppesen, The Style of Palestrina and the Dissonance, Dover Publications, INC ed., 2012.

[2] D. Conklin, "Music generation from statistical models," in *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Aberystwyth, Wales, 2003.

[3] D. Conklin and I. H. Witten, "Multiple viewpoint systems for music prediction," *Journal of New Music Research,* vol. 1, no. 24, pp. 51-73, 1995.

[4] D. Conklin, "Representation and discovery of vertical patterns in music," in *Music and Artificial Intelligence*, C. Anagnostopoulou, M. Ferrand and A. Smaill, Eds., Springer-Verlag Lecture Notes in Artificial Intelligence 2445, 2002, pp. 32-42.

[5] D. Conklin, "Chord sequence generation with semiotic patterns," *Journal of Mathematics and Music,* vol. 10(2), pp. 92-106, 2016.

[6] D. Conklin and M. Bergeron, "Discovery of contrapuntal patterns," in *ISMIR 2010:11th International Society for Music Information Retrieval Conference*, Utrecht, 2010.

[7] J. D. Fernandez and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research,* vol. 48, pp. 513-582, 2013.

[8] L. Hiller and L. Isaacson, "Musical composition with a high-speed digital computer," 1958.

[9] J. Rothgeb, Harmonizing the unFig.d bass: A computational Study., Thesis, Ph.D, Yale University ed., New Haven, Connecticut, 1968.

[10] K. Ebcioğlu, "Computer Counterpoint," in *International Computer Music Conference*, New York, 1980.

[11] K. Ebcioğlu, "An Expert System for Harmonizing Four-part Chorales," *Computer Music Journal,* vol. 12, no. 3, pp. 43-51, 1988.

[12] C. P. Tsang and M. Aitken, "Harmonizing music as a discipline of constraint logic programming," in *Proceedings of the International Computer Music Conference ICMC*, Montreal, Canada, 1991.

[13] R. Ovans and R. Davison, "An interactive Constraint-Based expert assistant for music composition," in *Proceedings of the Canadian Conference on Artificial Intelligence*, Ottawa, Canada, 1992.

[14] R. Ramirez and J. Peralta, "A constraint-based melody harmonizer," in *ECAI'98 Workshop on Constraints and Artistic Applications*, Brighton, 1998.

[15] S. Phon-Amnuaisuk, A. Tuson and G. Wiggins, "Evolving musical harmonisation," in *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, Portorož, Slovenia, 1999.

[16] D. Herremans and K. Sörensen, "Composing fifth species counterpoint music with a variable neighborhood search algorithm," *Expert Systems with Applications,* vol. 40, no. 16, pp. 6427-6437, 2013.

[17] M. Komosinski and P. Szachewicz, "Automatic species counterpoint composition by means of the dominance relation," *Journal of Mathematics and Music,* vol. 9, no. 1, pp. 75-94, 2015.

[18] A. Yilmaz and Z. Telatar, "Note-against-note two-voice counterpoint by means of fuzzy logic," *Knowledge-Based Systems,* vol. 23, no. 3, pp. 256-266, 2010.

[19] C. Roads, "Grammars as representations for music," *Computer Music Journal,* vol. 3(1), pp. 48-55, 1979.

[20] S. R. Holtzman, "Using generative grammars for music composition," *Computer Music Journal,* vol. 5(1), pp. 51-64, 1981.

[21] K. Jones, "A space grammar for the stochastic generation of Multi-Dimensional structures," in *International Computer Music Conference*, San Francisco, USA, 1980.

[22] R. A. McIntyre , "Bach in a box: the evolution of four part baroque harmony using the genetic algorithm," in *IEEE Conference on Evolutionary Computation*, 1994.

[23] A. Horner and L. Ayers, "Harmonization of musical progressions with genetic algorithms," in *Proceedings of the International Computer Music Conference*, 1995.

[24] S. Phon-Amnuaisuk, "Control language for harmonisation process," in *International Conference on Music and Artificial Intelligence*, Edinburgh, Scotland, UK, 2002.

[25] J. Polito, J. M. Daida and T. F. Bersano Begey, "Musica ex machina: Composing 16th-Century counterpoint with genetic programming and symbiosis," in *Proceedings of the International Conference on Evolutionary Programming*, 1997.

[26] N. Gwee, Complexity and Heuristics in Rule-Based Algorithmic Music Composition, Ph.D. thesis, Louisiana State University, 2002.

[27] P. Cruz-Alcázar and E. Vidal-Ruiz, "Learning regular grammars to model musical style: Comparing different coding schemes," in *International Colloquium on Grammatical Inference*, 1998.

[28] E. Gilbert and D. Conklin, "A probabilistic context-free grammar for

melodic reduction," in *In international Workshop on Artificial Intelligence and Music, at IJCAI-07, The Twentieth International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.

[29] R. Groves, "Towards the Generation of Melodic Structure," in *4th International Workshop on Musical Metacreation (MUME 2016). Held at the Seventh International Conference on Computational Creativity, ICCC 2016*, París, 2016.

[30] D. Quick and P. Hudak, "Grammar-based automated music composition in Haskell," in *In Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design*, Boston, Massachusetts, USA, 2013.

[31] M. Farbood, and B. Schoner, "Analysis and synthesis of Palestrina-style counterpoint using Markov chains," in *Proceedings of the International Computer Music Conference*, Cuba, 2001.

[32] D. Herremans, K. Sörensen and D. Conklin, "Sampling the extrema from statistical models of music with variable neighbourhood search," in *Proceedings of the Sound and Music Computing Conference*, Athens, Greece, 2014.

[33] M. Allan and C. Williams, "Harmonising chorales by probabilistic inferences," *Advances in Neural Information Processing Systems,* vol. 17, pp. 25-32, 2004.

[34] R. P. Whorley and D. Conklin, "Music Generation from Statistical Models of Harmony," *Journal of New Music Research,* vol. 45, no. 2, pp. 160-183, 2016.

[35] H. Hild, J. Feulner and W. Menzel, "HARMONET: A neural net for harmonizing chorales in the style of JS Bach," *Advances in Neural Information Processing,* p. 267–274, 1992.

[36] J. Feulner and D. Hörnel, "MELONET: neural networks that learn harmony-based melodic variations," in *Proceedings of the International Computer Music Conference*, San Francisco, 1994

[37] D. Hörnel and P. Degenhardt, "A neural organist improvising baroque-style melodic variations," in *Proceedings of the International Computer Music Conference*, Thessaloniki, Hellas, 1997.

[38] D. Hörnel and P. Degenhardt, "A neural organist improvising baroque-style melodic variations," in *Proceedings of the International Computer Music Conference*, Thessaloniki, Hellas, 1997.

[39] K. Verbeurgt, M. Fayer and M. Dinolfo, "A hybrid Neural-Markov approach for learning to compose music by example," in *Proceedings of the Canadian Conference on Advances in Artificial Intelligence*, 2004.

[40] K. Adiloglu and F. N. Alpaslan, "A machine learning approach to two-voice counterpoint composition," *Knowledge-Based Systems,* vol. 20(3), pp. 300-309, 2007.

[41] T. Feynman T, L and M. Gotham, M. Johnson and J. Shotto, "Automatic Stylistic Composition of Bach Chorales with Deep LSTM," in *ISMIR*, 2017.

[42] D. Precup and Y.W. Teh, "DeepBach: a Steerable Model for Bach Chorales Generation," *Proceedings of the 34th International Conference on Machine Learning*, 2017. pp. 1362-1371, 2007.

[43] I. Knopke and F. Jürgensen, "A system for identifying common melodic phrases in the masses of Palestrina," *Journal of New Music Research,* vol. 38, no. 2, pp. 171-181, 2009.

[44] K. A. Sidorov, A. Jones and A. D. Marshall, "Music Analysis as a Smallest Grammar Problem," in *In Proceedings of the 15th International Society for Music Information Retrieval (ISMIR)*, Canada: International Society for Music Information Retrieval, 2014.

[45] I. Khan and A. Jain, "A Comprehensive Survey on Sequential Pattern Mining," *International Journal of Engineering Research and Technology,* vol. 1, no. 4, June 2012.

[46] J. Wang and J. Han, " BIDE: Efficient mining of frequent closed sequences," in *Proceedings. 20th International Conference on Data Engineering*, Boston, Massachusetts, USA , 2004.

[47] F. Pachet, P. Roy and G. Barbieri, "Finite-Length Markov Processes with Constraints," in *In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Spain, 2011.

[48] T. Collins, R. Laney, A. Willis and P. H. Garthwaite, "Developing and Evaluating Computational Models of Musical Style," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 30 (1), pp. 16-43, 2016.

[49] D. Cope, "Experiments in musical intelligence (EMI): Non-linear linguistic-based composition," *Journal of New Music Research,* Vols. 18(1-2), pp. 117-139, 1989.

[50] D. Cope, Virtual music: computer synthesis of musical style, MIT Press, Cambridge, 2001.

[51] D. De la Motte, Contrapunto, S.A. Idea Books ed., Barcelona, 1981, p. 420.

[52] M. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data", *in Proceeding of International Society for Music Information Retrieval.* 2010.

[53] M. Zaki, "SPADE, An efficient algorithm for mining frequent sequences," *Machine learning,* Vols. 42(1-2), pp. 31-60, 2001.

[54] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proceedings of the 17th International Conference on Data Engineering*, Washington, DC, USA, 2001.

[55] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *International Conference on Extending Database Technology*, 1996.

[56] X. Yan, J. Han and R. Afshar, "CloSpan: Mining: Closed Sequential Patterns in Large Datasets," in *Proceedings of the 2003 SIAM International Conference on Data Mining*, San Francisco, California, USA, 2003.

[57] C. Li and J. Wang, "Efficiently Mining Closed Subsequences with Gap Constraints," in *SIAM International Conference on Data Mining*, Atlanta, Georgia, USA, 2008.

[58] J. Ayres, J. Gehrke, T. Yiu and J. Flannick, "Sequential Pattern Mining using A Bitmap Representation," in *Proceedings of the International Conference on Knowledge*, Edmonton, Canada, 2002.

[59] R. Gjerdingen, "Concrete musical knowledge and a computer program for species counterpoint," in *Explorations in Music, the Arts, and Ideas: Essays in Honor of Leonard B. Meyer*, Pendragon Press, 1988, pp. 199-220.

[60] K. Jeppesen and H. Glen, Counterpoint: the Polyphonic Vocal Style of the Sixteenth Century, Englewood Cliffs, N. J., Prentice-Hall, 1960, p. 330.

[61] V. Padilla and D. Conklin, "Statistical generation of two voice florid counterpoint," in *Proceedings of the International Conference on Sound and Music Computing*, Hamburg, Germany (2016).

Víctor Padilla Martín-Caro

Dr. Víctor Padilla Martín-Caro was born in Madrid, Spain. Currently, he is the Vice-Dean of Social Sciences and Humanities at UNIR, Universidad Internacional de La Rioja. Previously, he worked as a musical research at Lancaster Institute for the Contemporary Art at Lancaster University, UK. His research interests range from musical composition, optical music recognition (OMR), to music information research (MIR) for pattern discovery.

Darrell Conklin

Prof. Darrell Conklin is an Ikerbasque Research Professor at the Department of Computer Science and Artificial Intelligence at the University of the Basque Country, San Sebastian, Spain and the group leader of the Music Informatics Group. His research areas include music generation, statistical modelling, and pattern discovery in music.