STATISTICAL GENERATION OF TWO-VOICE FLORID COUNTERPOINT

Víctor Padilla Martín-Caro[1] **Darrell Conklin**[1, 2]
[1]Department of Computer Science and Artificial Intelligence,
University of the Basque Country UPV/EHU, San Sebastián, Spain
[2]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
victor.padilla.mc@gmail.com darrell.conklin@ehu.eus

ABSTRACT

In this paper, we explore a method for statistical generation of music based on the style of Palestrina. First, we find patterns in one piece that are selected and organized according to a probabilistic distribution, using horizontal viewpoints to describe melodic properties of events. Once the template is chosen and covered with patterns, two-voice counterpoint in a florid style is generated using a first-order Markov model with constraints obtained from the template. For constructing the model, vertical slices of pitch and rhythm are compiled from a corpus of 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.

1. INTRODUCTION

In 1725, Johann Joseph Fux presented his “Gradus Ad Parnassum”, a pedagogical method that breaks the learning task into well-defined graduated stages, from note against note through to florid counterpoint. This method continues to be a standard counterpoint text studied by a huge number of musicians. However, generating music based on rules is not a good stylistic approach: in fact, music from Renaissance to Romanticism can be written following basically the same rules. For example, stylistic differences between the Bach and Palestrina counterpoint is not defined by basic generic rules (parallel fifths or octaves, e.g.), and implementing specific constraints and exception to them is a very complex task. In other words, musical style has to be learnt from examples in order to model as closely as possible the composer's style. Tuning rules and exceptions can be exhausting and the rule-based system achieved becomes, in many cases, imprecise.

The option we present here is is to construct models with the real pieces of the composer. The corpus of pieces we are working with comprises 717 movements from Palestrina masses, providing a huge amount data for training statistical models of the Renaissance style.

These data are closely related with the music style without hard coding rules, but to compose a piece of music in the Renaissance style is not so simple. Counterpoint is full of imitations, canons, motifs, augmentations, etc. and such devices are not easily captured by a first-order Markov model [14]. For solving these limitations and to provide coherence to the generated pieces, we take a piece from the corpus and obtain the main patterns based on different viewpoints, as will be outlined in Section 3. The patterns are used to cover the template piece. In generated music the rhythm of the template piece is retained.

2. PRIOR WORK

The generation of new music based on rules, or expert systems, has a long tradition, from the works in 1955 of Hiller and Isaacson [1] using the ILLIAC computer at the University of Illinois. More recent work is the article of Ebcioglu implementing rules for counterpoint and Bach chorales [2]. Herremans and Sorensen work with different counterpoint species using a variable neighborhood search algorithm [3]. Komosinski and Szachewicz [4] address the difficulty of evaluating penalty (or reward) values for each broken (or satisfied) rule. A simple additive rule weighting function is weak and they propose to use the dominance relation. Implementing rules by fuzzy logic to generate two-voice first species counterpoint is analyzed by Yilmaz and Telatar [5].

On the other hand, research that uses machine learning technologies has been increasing in recent years. One possible way is to work with neural networks [6] that learn from the context developing a Bach-style choral harmonization. The probabilistic approach (HMM) has been studied by Allan and Williams using a Bach chorale corpus [7]. Focused on counterpoint generation and HMM, Farbood and Schoner [8] work with first-species counterpoint. The method uses Markov chains which capture the rules of counterpoint using probabilistic tables for harmony, melody, parallel motion, and cadences.

David Cope’s Experiments in Musical Intelligence (EMI) is a system of algorithmic composition created and developed since 1981. His approach is based on Transition Networks analogous to the historical model of the musical dice game, but detecting and deciding the components autonomously. The semantic classification of the material is carried out by a system called SPEAC [9] (statement, preparation, extensions, antecedent and consequent). The SPEAC system is inspired by the analysis method developed by Heinrich Schenker.

Regarding the detection of melodic phrases in the masses of Palestrina, Knopke and Jurgensen [10] describe a system based on the use of suffix arrays to find repeated patterns. The main drawback of their system is the requirement for exact matching of the patterns. Augmentations, diminutions or non-exact intervals (fourth by fifth or third minor by major) are not considered.

Many sequential pattern mining methods have been developed in the last decade, such as SPADE, PrefixSpan, GSP, CloSpan or BIDE [11]. In our research, for analysing patterns, we are using a gap-BIDE [12] algorithm with zero gaps between sequences as will be explained in 4.2.

For slicing and obtaining patterns from scores, we are using the concept of horizontal and vertical viewpoints. This idea of viewpoints was developed and refined by Conklin [13,14,15,16] and will be described in Section 4.

This paper is organized as follows. Section 3 justifies the corpus chosen. Section 4 develops the idea of horizontal and vertical viewpoints and the algorithm to find patterns. That section also explores the concept of probability with respect to zero- and first-order Markov models of the Palestrina corpus. Section 5 explains the restrictions imposed by the template and the results obtained.

3. PALESTRINA'S MASSES

The corpus of pieces we are working with consists of 101 masses composed by Palestrina. 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 number of voices from three to six. For example, Benedictus in many masses is written in three voices and Kyrie in five or six. Below is the corpus of pieces we have, using the data of music21[1], which is a Python-based toolkit for computer-aided musicology developed by MIT.

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

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 this style as basic training for new students in composition. Another important aspect is the homogeneity of the corpus of pieces. There are not significant differences in style between the first and the last mass, and the number of pieces is big enough to build a probabilistic model. Regarding this point and taking into account just two voices, the number of vertical slices available is almost 350,000 which provide enough information for constructing a reasonably accurate first-order Markov model, as is explained in the next section.

4. VIEWPOINTS FOR PATTERN DISCOVERY

For generation of polyphony, both horizontal (melodic) and vertical (harmonic) aspects must be modelled. In this work, we implement the concept of *linked viewpoints*, from the horizontal and vertical perspective. As an overview [15], a viewpoint system is a collection of independent views of the musical surface each of which models a specific type of musical phenomena. A piece of music is therefore transformed into a higher level description derived from the basic surface representation. For every viewpoint a *viewpoint sequence* function transforms a sequence of basic events into a sequence of defined viewpoint elements. A linked viewpoint is a combination of two or more viewpoints that models other viewpoints simultaneously.

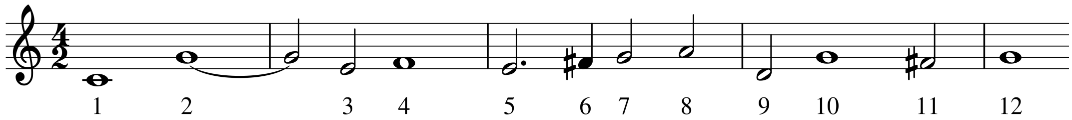
4.1 Horizontal viewpoints

Each phrase of Palestrina music is treated as a sequence of linked viewpoints. 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 (Figure 1), for example, absolute pitch, name of note (spell), melodic contour, duration contour, interval or a group of interval joined (scalestep). A pattern is a sequence of features $(v_1, \ldots, v_l)$ where each v_i is a feature (e.g. scale step linked with contour of duration, as specified in Equation 1).

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)

[1]http://web.mit.edu/music21/



	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

Figure 1. Different viewpoints from a melody of Palestrina.

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 Figure 2. Using this scalestep viewpoint, equivalent intervals, from the point of view of imitation, are grouped into the same category.

4.2 Finding patterns

Each file from the corpus is converted to a viewpoint sequence, separating phrases by rests. A link between-viewpoints is defined using the constructor ⊗. The linked viewpoint for discovering patterns is:

$$\text{scalestep} \otimes \text{contour(duration)} \quad (1)$$

This particular linking of viewpoints allows the discovery of augmented and diminished patterns and melodic inversions.

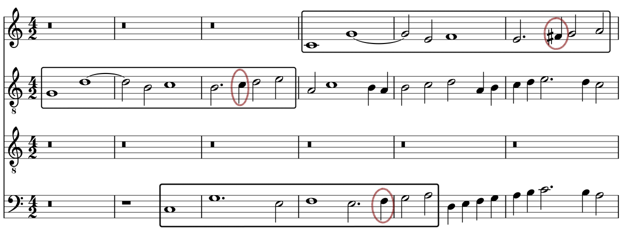


Figure 2. Agnus from *Beata Marie Virginis*. Palestrina. Patterns with different intervals.

Data mining [11] is the computational process of discovering patterns in a large data set. 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, pattern discovery in protein sequences, etc. A sequence of musical features can be analysed as a sequence of DNA and, taking ideas from biology, to identify similar and repeated patterns through the string of elements.

Well known algorithms for sequential pattern mining [10] are SPADE (Sequential PAttern Discovery using Equivalence classes), PrefixSpan (Prefix-projected Sequential pattern mining), GSP (Generalized Sequential Pattern algorithm), CloSpan (Closed Sequential pattern mining) or BIDE (BI-Directional Extension). In our experiments we are using gap-BIDE, 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.

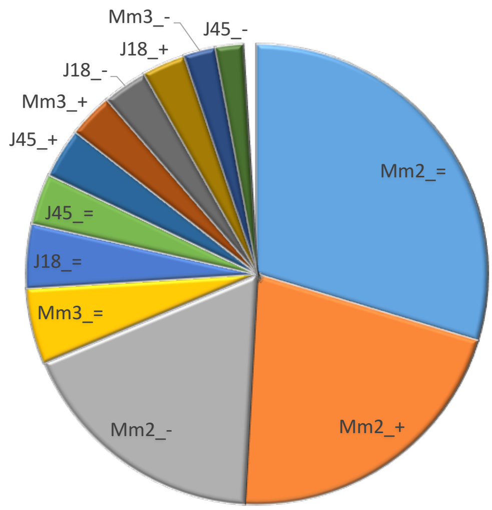


Figure 3. Distribution of different features in the corpus using Equation (1)

4.3 Ranking patterns

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.

1: 30.74	1	1	1	1	1	1	1	4	91	95	160
2: 30.19	2	2	2	2	2	2	2	5	95	160	
4: 27.08	4	4	4	4	4	4	4	11	160: 0.00		
5: 26.64	5	5	5	5	5	5	5	24			
7: 23.35	7	7	7	9	10	10	10	25			
9: 21.60	9	9	9	10	11	11	11	39			
12: 19.28	10: 19.72	10	10	11	24	24	24	69			
	11: 19.60	11	11	20	25	25	25	91			
	12	12	12	24	39	39	39	95: 2.17			
	20: 13.82	20	20	25: 11.31	41	41	41	131			
	36: 8.69	24: 11.72	24	39	42	42	42				
		36	36	41	52	52	69				
		41: 8.04	39: 8.38	42: 7.89	69: 4.58	69	77				
		52: 6.48	41	52	77	77	91: 2.38				
		120: 1.00	52	77	99	99	131				
			77: 3.46	99: 1.96	111	111	143				
			111: 1.41	111	143: 0.11	131: 0.48					

Figure 4. Agnus from *Beata Marie Virginis* of Palestrina (Agnus.krn). Palestrina. Different patterns ordered by binomial distribution. The first number gives the rank of the pattern, and the second the $\mathbb{I}$ value for the first position of a pattern.

Figure 3 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 (30.0%), higher duration the second note (21.4%) and lower duration the second note (18.0%). The rest of the 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.

One possible way of coding the patterns is as follows (Table 1). The odd lines (grey) indicate the array of linked features and the even ones (white) the position of the pattern in different phrases. The three numbers specifying the position are the indices of the phrase, start note and end note of the pattern.

Patterns
<J45_+> <Mm3_-> <Mm2_+> <Mm2_-> <Mm2_-> <Mm2_+> <Mm2_=>
(4, 0, 7), (13, 0, 7), (27, 0, 7)
<J45_+> <Mm3_-> <Mm2_+>
(0, 0, 3), (4, 0, 3), (13, 0, 3), (21, 0, 3), (27, 0, 3)
<J45_+> <Mm3_-> <Mm2_+> <Mm2_+> <Mm2_-> <Mm2_+>
(0, 0, 6), (21, 0, 6)

Table 1. Example of coding patterns. Agnus from *Beata Marie Virginis* of Palestrina.

The background probability (b_p) of a pattern $p = (v_1, \dots, v_l)$ must be estimated, for example using a zero-order model of the corpus

$$b_p = \prod_{i=1}^{l} c(v_i)/N \quad (2)$$

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 $\mathbb{I}$ can be defined using the binomial distribution, which gives the probability of finding at least the observed number of occurrences of the pattern.

$$\mathbb{I}(p) = -\ln \mathbb{B}_{\geq}(k_p; t_p, b_p) \quad (3)$$

where:

- $\mathbb{B}_{\geq}$ gives the cumulative probability (right tail) of the binomial distribution,
- t_p approximates the maximum number of positions that can be possibly matched by the pattern,
- k_p is the number of times the pattern appears in the template piece.

Figure 4 is one example of different patterns found in one fragment of Agnus from *Beata Marie Virginis* of Palestrina, ordered by (3). The number followed by a colon (:) indicates the $\mathbb{I}$ for each pattern. This template will be used for creating the new piece as is explained in the next section.

4.4 Vertical viewpoints. Markov model

For constructing the Markov model, two voices are selected and cut into slices (see Figure 5). In this first approach, we have taken the highest and lowest voice for a better result, removing the intermediate. Usually, the music that follows harmonic constraints entrust to the lower part (bass) an important role in the harmonic context, while the higher (soprano) is more appropriate for defining melodies.

The slicing process is the same as the method explained by Conklin [17], dividing when a new event appears in one voice. In our example, for simplicity, we do not retain continuations or ties (the full expansion method of [17]). In Renaissance vocal music, the repetitions or ties sometimes depend on the text and, in the new score, the durations are going to be obtained from the template.

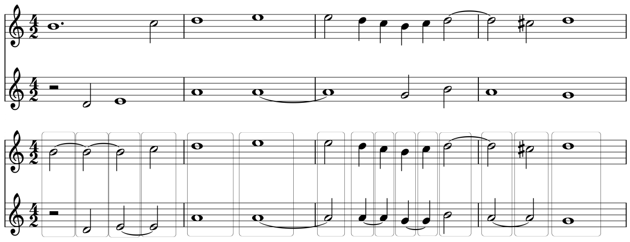


Figure 5. Example of two-voice slicing.

Taking into account pitch and duration, the number of slices 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 Figure 6. The vertical axis is the number of repetitions (logarithmic scale) and the horizontal the slice ordered by repetitions. Counting the number of unique next slices (first-order Markov model), ordered by the zero-order model, the results are shown in Figure 7, where the number of different paths ranges from 0 and 183.

Figure 6. Zero order distribution of repetitions.

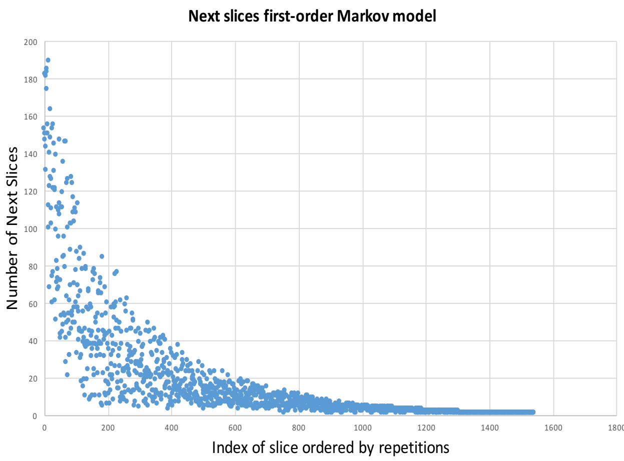


Figure 7. Distribution of unique next slices, first-order Markov model.

Figure 8. Generation of upper voice based on different constraints in lower voice.

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 this melody as a pattern (Figure 8), 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.

5. APPLYING THE MODEL TO THE TEMPLATE

This section explains a method for generating new music based on a piece from the corpus. The idea is to take the patterns of this piece, which guarantees coherence, and fill the template with the slices and probabilistic paths obtained by the first-order model. The steps are as described below.

5.1 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 we have all the possible patterns, a greedy algorithm is used for covering the score. This algorithm takes the better patterns from left to right. For practical purposes, in the case of simultaneous patterns in two voices, they are evaluated removing the item that has a lower score. The template is, therefore, divided into different regions that are separated based on the patterns. Figure 9 shows the first six measures of the Agnus from

Beata Marie Virginis and how the patterns are found in the different voices. For simplicity, we will take the upper and lower voice for the generation.

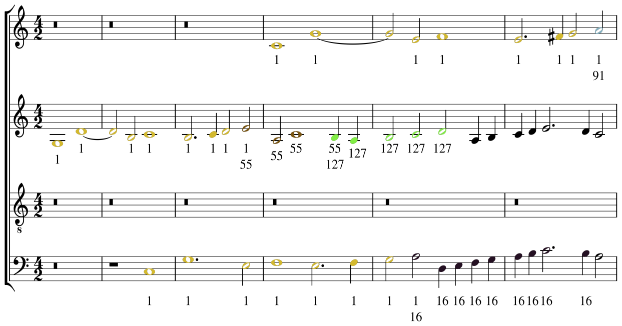


Figure 9. Agnus from *Beata Marie Virginis* of Palestrina. Covering the piece using a greedy algorithm.

Once the final template is decided, for constraining notes within areas covered by patterns, the viewpoint

$$\text{scalestep}\otimes\text{duration}\otimes\text{contour(pitch)} \qquad (4)$$

is used. Note that this represents a slightly more restrictive pattern 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 paper the exact durations from the Palestrina template are used, therefore the conservation of duration is assured.

Regarding the corpus of pieces, some of them are composed of just three voices (e.g. Benedictus). For testing purposes, we take Benedictus from the mass *Descendit angelus Domini* as a template and proceed with the next steps:

- First, 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 Equation (4) is used for horizontal restrictions. If the region is not a pattern, just the duration viewpoint remains.
- The vertical slices are filtered by the different constraints. If at one point it is not possible to find next slice, a backtracking algorithm is performed (see Section 5.2).

There is a probability associated with each new slice in the first-order Markov model. Different results will be obtained choosing a different range of probabilities, as will be commented in Section 6.

5.2 Double 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 possibilities 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.

Figure 10 shows one piece generated by this method. The two upper systems are the original Benedictus from *Descendit angelus Domini*. The colors indicate the patterns found by the greedy algorithm. The three lower systems are the music generated based on the upper and lower voice from the template.

6. CONCLUSIONS AND NEXT STEPS

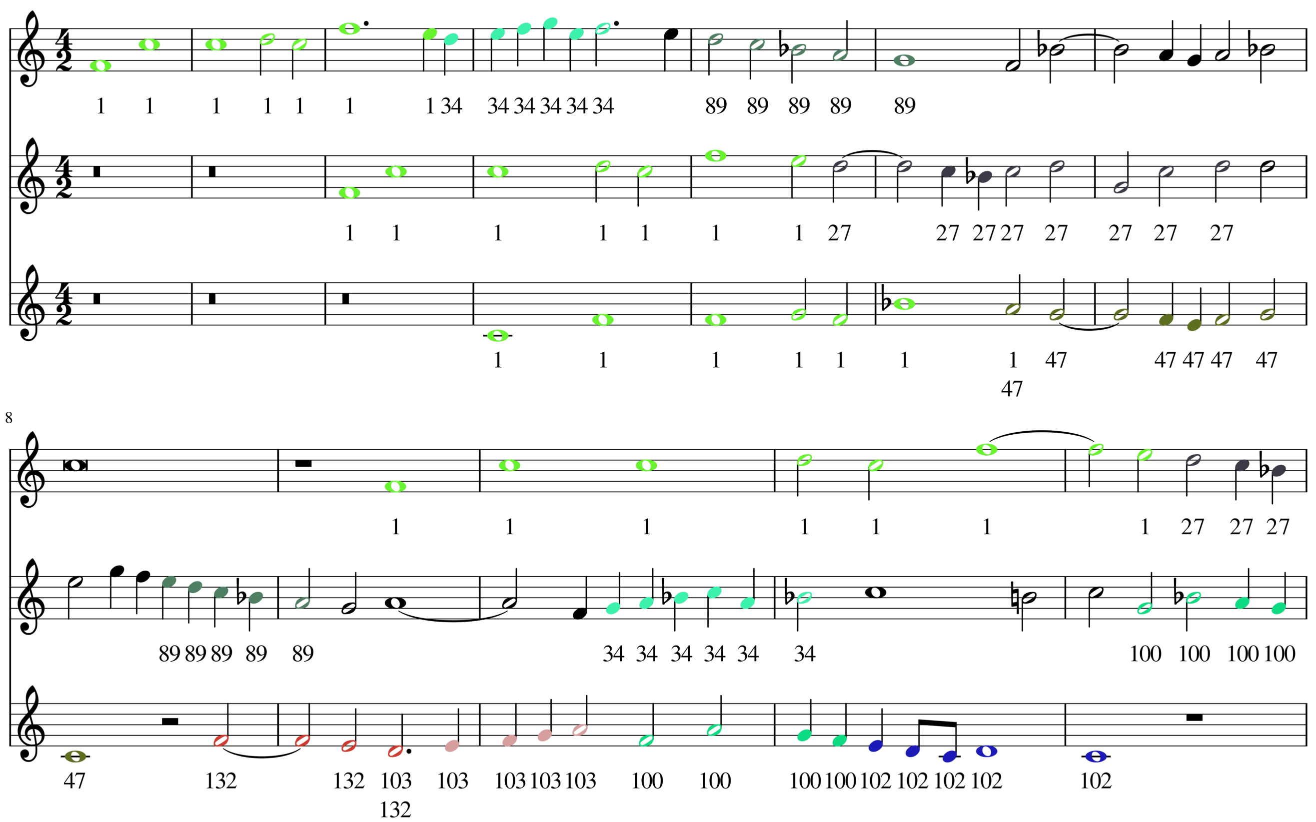
This paper presents a method for generating new music based on the corpus of masses of Palestrina. The system, for practical purposes, is limited to two voices taking a template from the corpus without overlapping patterns.

The gap-BIDE algorithm and the binomial distribution explained in Section 4.2 works correctly in most of the cases, and the ranking of the patterns discovered 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 template extracted should be done.

Regarding the Markov model, the zero- and first-order are a good approach for ensuring correctly linked slices with rhythm and pitch constraints. The first-order Markov model prevents “weak” linked chords with grammatical errors such as parallel fifths, parallel octaves, without implementing specific rules. This model does not organize harmonic regions, and “non-idiomatic” melodic movements can appear. 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, (David Cope’s EMI, e.g.), the coherence is achieved analyzing bigger slices of the pieces. 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 5.2 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

Benedictus from *Descendit angelus Domini*. Template from the original score



New music generated based on the upper and lower voice from the previous template.

Figure 10. Template and new music generated. Benedictus from *Descendit angelus Domini* of Palestrina (Benedictus_19.krn). The upper score is the original from Palestrina. The lower score is two voices generated based on higher and lower part. The colors identify different patterns.

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, 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.

Acknowledgments
This research is 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.

7. REFERENCES

[1] L. Hiller and L. Isaacson, "Musical composition with a high-speed digital computer," *Machines Models of Music*, pp. 9-21, 1958, Reprint of original article in Journal of Audio Engineering Society.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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*, pp. 267–274, 1992.

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

[8] 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, pp. 471-474.

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

[10] 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.

[11] 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.

[12] Li Chun and Wang Jianyong, "Efficiently Mining Closed Subsequences with Gap Constraints," in *SIAM*, 2008, pp. 313-322.

[13] D. Conklin, "Discovery of distinctive patterns in music," *Intelligent Data Analysis,* vol. 14, no. 5, pp. 547-554, 2010.

[14] 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, pp. 30-35.

[15] 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.

[16] D. Conklin and C. Anagnostopoulou, "Representation and Discovery of Multiple Viewpoint Patterns," in *Proceedings of the International Computer Music Conference*, San Francisco, 2001, pp. 479-485.

[17] 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.

[18] M. Bergeron and D. Conklin, "Temporal patterns in polyphony," in *Mathematics and Computation in Music*. Berlin, Heidelberg: Springer, 2009, pp. 32-42.

[19] M. Bergeron and D. Conklin, "Subsumption of vertical viewpoint patterns," in *Mathematics and Computation in Music*, MCM 2011 Third International Conference, Ed. Paris, France, Springer, 2011, pp. 1-12.

[20] D. Conklin and M. Bergeron, "Feature set patterns in music," *Computer Music Journal*, vol. 32, no. 1, pp. 60-70, 2008.