

Musical Style Modification as an Optimization Problem

Frank Zalkow

Saarland University

frank.zalkow@uni-saarland.de

Stephan Brand

SAP SE

stephan.brand@sap.com

Benjamin Graf

SAP SE

benjamin.graf@sap.com

ABSTRACT

This paper concerns musical style modification on symbolic level. It introduces a new, flexible method for changing a given piece of music so that its style is modified to another one that previously has been learned from a corpus of musical pieces. Mainly this is an optimization task with the music's note events being optimized for different objectives relating to that corpus. The method has been developed for the use case of pushing existing monophonic pieces of music closer to the style of the outstanding electric bass guitar player Jaco Pastorius.

1. INTRODUCTION

Musical style is a quite vague term that is at risk not to be captured computationally or even analytically. The musicologist Guido Adler states in his early, grand monograph about musical style:

So [regarding the definition of style] one has to content oneself with periphrases. Style is the center of artistic approaching and conceiving, it proves itself, as Goethe says, as a source of knowledge about deep truth of life, rather than mere sensory observation and replication. [1, p. 5]¹

This passage suggests not to approach musical style computationally. Nearly 80 years after that Leonard B. Meyer expresses a rather opposite view:

Once a musical style has become part of the habit responses of composers, performers, and practiced listeners it may be regarded as a complex system of probabilities. That musical styles are internalized probability systems is demonstrated by the rules of musical grammar and syntax found in textbooks on harmony, counterpoint, and theory in general. [...] For example, we are told that in the tonal harmonic system of Western music the tonic chord is most often followed by the dominant, frequently by the subdominant, sometimes by the submediant, and so forth. [2, p. 414]

There are a couple of things to notice here: First, Meyer supports the view that style isn't in the music *per se*, but only when regarded in relation with other systems. Second,

¹ Translation by the author. Original version:

So muß man sich mit Umschreibungen begnügen. Der Stil ist das Zentrum künstlerischer Behandlung und Erfassung, er erweist sich, wie Goethe sagt, als eine Erkenntnisquelle von viel tieferer Lebenswahrheit, als die bloße sinnliche Beobachtung und Nachbildung.

Copyright: ©2016 Frank Zalkow et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

style is seen probabilistic, supporting the attempt of this paper to tackle style computationally. Third, in the textbooks he mentions, probabilities are used in a very broad sense. Words like *frequently* or *sometimes* aren't enough for the models of this paper. So one cannot rely on textbooks and has to work through real data.

Working through data is the guidance of this new approach for musical style modification, i.e. changing a given piece of music so that its style is modified to another one that previously has been learned from a data corpus, while the original piece of music should shine through the new one (section 2). This style modification is seen as an optimization problem, where the music is to be optimized regarding different objectives (section 4). The method has been developed for monophonic melodic bass lines, along with chord annotations—especially for the style of the outstanding electric bass guitar player Jaco Pastorius (see section 5 for some results). Due to the nature of the use case, the method is oriented towards monophonic symbolic music, but most parts of the procedure could easily be extended to polyphonic music as well. In this case the chord annotations may be even computed in a preceding automated step, so that annotating would not be necessary.

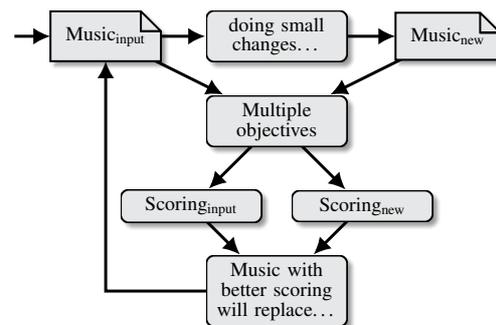


Figure 1: Rough overview of the modification procedure

2. STYLE MODIFICATION PROCEDURE

A simple monophonic, symbolic music representation can be seen as a series of I note events N , where each note event n_i is a tuple of pitch p_i as MIDI note number and duration d_i in quarter lengths (ql).

$$N = (n_1, n_2, \dots, n_I) \text{ where } n_i = (p_i, d_i) \quad (1)$$

In the case of rest, a predefined rest symbol takes the place of note number. Along with the note events a chord annotation is needed, also being represented as a series of J chord events C , where each chord event c_j is a tuple of chord symbol s_j and duration d_j , again in ql.

$$C = (c_1, c_2, \dots, c_J) \text{ where } c_j = (s_j, d_j) \quad (2)$$

Such an existing N with a corresponding C should be modified in such a way that it comes closer to a specific style. C is fixed and won't be changed. This task is viewed as a local search, performing a multi-objective optimization for finding a local-optimal N [3]. The main idea is to start from a piece of music and try out *neighbors*. If a neighbor is better than the original one, according to the objectives described in section 4, the neighbor is saved and the process is iteratively continued with this new one. In a multi-objective optimization it is not straightforward what is considered *better*. For ensuring that no objective becomes worse, Pareto optimality is utilized: A state is only considered better, if all objectives stay the same or increase in value. A neighbor is reached by randomly doing one of the following changes:

- Changing the pitch p_i of a single note event. The maximum change interval is a major third upwards or downwards.
- Changing the duration of two notes d_{i_1} and d_{i_2} so that the overall duration of the note succession in this voice stays the same.
- A note n_i is divided into multiple ones, having $(1/2, 1/2)$, $(2/3, 1/3)$ or $(1/3, 1/3, 1/3)$ of the original duration d_i .
- Two successive notes n_i and n_{i+1} are joined into a single one with duration $d_i + d_{i+1}$.

Ideally one wants to have a manageable amount of neighbors so that every one can be tried to find out which one is best. Unfortunately the amount is not manageable in this case: For a given state with N note events, there are $16N^2 - 4N - 1$ possible neighbors (if one assumes that each possibility of splitting/joining notes is always valid). Because one cannot try out all possibilities, one randomly tries out neighbors and applies a technique from the toolbox of metaheuristics: Tabu search, which means storing the already tried out possibilities for not visiting them again. But even by this only a small fraction of possibilities will be tried out and one will be trapped in a local optimum. In most cases many efforts are employed to overcome local optima for finding the global one or at least solutions better than the first local optimum. But here there is a relaxing property: There is no need for finding the global optimum anyway. Changing a given piece of music for pushing it into a specific style should not mean to completely throw away the original piece and replace it by the stylistic *best* piece ever possible. It is reasonable doing small modifications until no modification improves the values of all objectives. By that it can be assured that the modified piece retains characteristics of the original one.

3. DATA CORPUS

Before formulating objectives, a data corpus of examples of music in the target style as well as counterexamples has to be established. The choice of counterexamples is crucial: Ideally a musical corpus would be needed that covers all music ever possible that doesn't belong to the target style in a statistically significant way. In practice this isn't possible. As an compromise, on the one hand it should be a style quite different from the target style, on the other hand it should be not too far off, so that the target style is sharpened by differentiating it from the counterexamples.

In our use case pieces of Jaco Pastorius (8 pieces of 2227.5 ql duration in total) form the target corpus whereas pieces of Victor Wooten (8 pieces of 3642.75 ql duration in total) form the counterexamples. So both corpora contain electric bass guitar music in the genre of jazz rock, whereas both bassists clearly show a different style. The complete list of pieces used is to be found in [4].

In the following section the objectives used in the Pastorius-project are introduced. But a key feature of this method is its flexibility: If some of its objectives don't seem appropriate or other objectives seem to be needed for a specific target style, it is easy to leave some out and/or develop new ones.

4. OBJECTIVES

4.1 Feature Classification

Before tackling the modification task, a classification task is to be solved. For that purpose one has to design individual feature extractors, tailored to the specifics of the target style. If, like in our case, it is to be assumed that style is not only recognizable when the full piece has been played, but also on a local level, one can apply windowing of a fixed musical duration.

The feature extractors compute for each window a row vector x_i , forming the feature matrix X . A target value y_i can be assigned to each feature vector, numerically representing the target style with 0 and the style of the counterexamples with 1. So for the corpus the target values are obvious, forming the column vector Y . Therefore a function $f : X \rightarrow Y$ is needed. This can be learned from the data corpus, where cross validation assures a certain generalizability. For learning this function we made good experiences with Gradient Tree Boosting, because of its good performance as well as the interpretability of the utilized decision trees. For details regarding Decision Trees and Gradient Tree Boosting see [5]. The Gradient Tree boosting classifier cannot only classify a given window of music, it also can report a probability of a the window belonging to the target style. This probability forms the first objective that is to be optimized.

This objective is the most flexible one because the feature design can be customized for the characteristics that the modified music should have. Adopting this method for new styles goes hand in hand with a considerable work in designing appropriate feature extractors.

Beside 324 feature dimensions coming from already implemented feature extractors of music21 [6], multiple feature extractors have been customly designed for the music of the Pastorius project, outputting 86 dimensions. As an example a feature extractor should be shortly described that aims to model one of the striking features of the bass guitar play of Jaco Pastorius, according Pastorius-expert Sean Malone:

Measure 47 [of *Donna Lee*, author's note] contains the first occurrence of what would become a Pastorius trademark: eighth-note triplets in four-note groups, outlining descending seventh-chord arpeggios. The effect is polyrhythmic – the feeling of two separate pulses within the bar that don't share an equal division. [...] As we will see, Jaco utilizes this same technique (including groupings of five) in many of his solos. [7, p. 6]

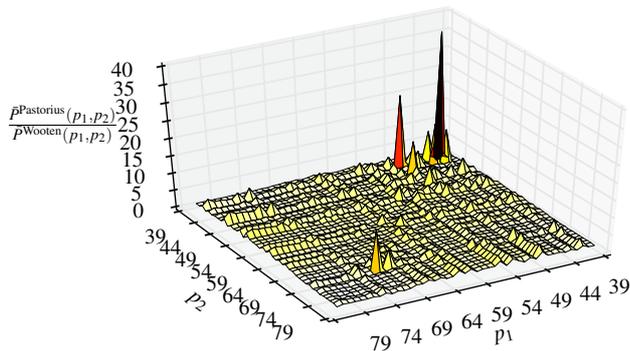


Figure 4: Surface of an objective function of two successive pitches regarding the ratio of the example/counter-example-ratio for the average smoothed Markov probabilities (order 0 and 1) with Jaco Pastorius being the target style and Victor Wooten being the counter-example.

4.4 Time correlations for chord-repetitions

The main idea of this objective is to capture some large scale structural similarities within the pieces of the target style. A shallow idea of large scale structure should also be given by the feature classification (subsection 4.1) when the relative position of the window is given as feature. But in practice large scale structure is something one misses most in the generated music. So this is an additional approach to include that. Our approach is based on two assumptions about repetitions in music: The first assumption is, that structure evolves by the absence or presence of repetition, e.g. the varied reoccurrence of material already played some time ago. The second assumption is that such kind of repetition most probably occurs when the relative changes in harmony also repeat. For each such a repetition both corresponding subparts of the note sequence N is converted into a piano roll representation (cf. figure 5c and d) where, after subtracting the mean and normalizing, a circular cross correlation (cf. figure 5e) is performed, so this objective correlates with motivic similarity. When referring to both parts as $N^{(1)}$ and $N^{(2)}$ with length I , the circular cross correlation is given by:

$$R_{N^{(1)}N^{(2)}}(l) = \frac{1}{I} \sum_{i=1}^I N_i^{(1)} N_{(i+l) \bmod I}^{(2)} \quad (6)$$

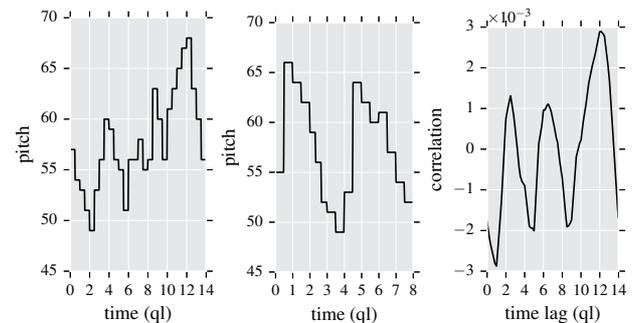
And that's how it is applied in the optimization: All cross-correlations between parts with related chord-progressions in the target style corpus are pre-computed. During the optimization the cross-correlations of those related sections are computed, too. Then the inner product between each correlation of the piece of music to be optimized and the ones of the target style corpus are computed and the maximum one is returned as value of this objective. Since the dot product of correlations of different lengths cannot be computed, all correlations of the same progression length are brought to the same length by linear interpolation. By that means one ensures that the correlation within the chord progressions in the music to be optimized becomes more similar to ones of the examples in the target style.



(a) from Pastorius' *Donna Lee*



(b) from Pastorius' *Donna Lee*



(c) Piano roll representation of a. (d) Piano roll representation of b. (e) Cross-correlation of c and d.



(f) Aligning of the two examples with b being circularly shifted to the maximum value of e.

Figure 5: Illustration of this objective for two examples with the same intervals between the roots of the chord progression (when enharmonic change is ignored). The cross-correlation shows that a circular shifting of 12 ql the two examples have the maximal motivic similarity.

5. RESULTS

In contrast to the classification, where the performance can be evaluated relatively impartially, such an objective evaluation is not possible for the modification. One can only try it out and evaluate the result subjectively. So the assessment heavily depends on the judges, their musical and cultural background, taste. As an example, a simple traditional melody has been chosen: *New Britain*, often sung along with the Christian hymn *Amazing Grace*. See figure 6 for the original version. During the modification process, 3398 different changes have been tried out whereby only 14 ones have been accepted by Pareto optimality. See figure 7 for the final version.



Figure 6: *New Britain* resp. *Amazing Grace*, original version.



Figure 7: *New Britain* resp. *Amazing Grace*, modified version.

The tying of notes from bars 4 to 5 seems to interrupt the arc of suspense, thus it seems to be rather unfavourable from a musical viewpoint. The run from bar 9 seems interesting and coherent. The two eighths in bar 9 as well as the ones in bar 15 seem to surround the following principal note, which lets the music appear quite natural. Especially the first example can be considered a broadened *double appoggiatura*. Some successions seem harsh, like the succession of the major to the minor third of the chord in bar 11 or the melodic succession of a semitone and a tritone in bars 12–13, but such harsh elements are not unusual in the music of Pastorius, see figure 8. Maybe they seem spurious, since one rather bears in mind the consonant original version, which is still noticeable in the modified version to a large extent. But from that point of view, the result is a successful blending of the original version and the style of Pastorius, even if it is doubtful if Pastorius would have improvised over *New Britain* like this.

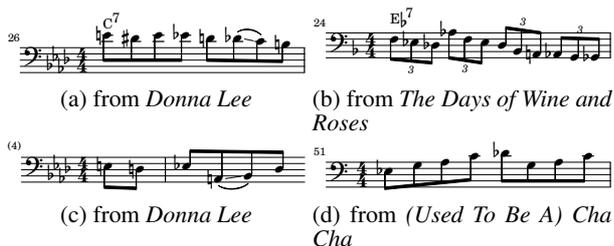


Figure 8: Jaco Pastorius examples, that could be the model for *harsh* results in the modification. a and b: Succession of the minor on the major third of the chord. c and d: Melodic succession of a semitone and a tritone.

6. RELATED WORK

One of the most prominent researchers, engaging computationally with musical style, especially in symbolic style synthesis, is David Cope [12–14]. In its basic form his style replication program EMI (“Experiments in Musical Intelligence”) has to be fed by over a thousand of user input questions. Cope also attempts to overcome this by automatically analyzing a corpus of music. Roughly, this involves finding what Cope calls signatures, frequently reoccurring sequences, assigning functional units to them and recombining the corpus with special regards to those functional signatures. This leads to impressive results for music with rather homogeneous texture, but it may be less appropriate for more eclectic and erratic styles, like the one

of Pastorius, because signatures are less dominant. Nevertheless, it would be interesting to apply Cope’s inspiring work to eclectic and erratic styles since Cope developed his methodology sophisticatedly, far exceeding the rough and basic ideas just touched here.

Cope describes basic categories into which music composing programs fall:

The approaches [...] include rules-based algorithms, data-driven-programming, genetic algorithms, neural networks, fuzzy logic, mathematical modeling, and sonification. Although there are other ways to program computers to compose music, these seven basic processes represent the most commonly used types. [12, p. 57]

If one would like to force the approach of this paper to fall into these categories, rules-based programming⁸ and data-driven-programming would fit, but a considerable amount of this work wouldn’t be described. Especially Cope’s category of *genetic algorithms* (GAs) is too specific and could be generalized to *metaheuristics*, which then would also fit for the method described here. GAs enable random jumps in the optimization neighborhood whereas the method presented here only takes small steps for ensuring that a local optimum is targeted—which is desired as described in section 2. Markov chains have a great tradition in music. They found application very early in both computer aided music generation [15–17] and in musicological studies [18, 19]. More recently, researchers from the Sony Computer Science Laboratory rediscovered Markov chains by combining them with constraint based programming, yielding very interesting results [10, 20]. In general, most music generation approaches, including Cope’s and all Markovian methods, are united by the strategy of recombining elements of an existing musical corpus. Other attempts that fall under this umbrella are suffix based methods [21, 22]. The method presented here, however, also enables recombinatorial results, but is not restricted to that because the other optimization objectives also foster the generation of music that is similar to the corpus on a more abstract level. [22, 23] share the concept of an underlying harmonic progression with the approach presented here. [24] is similar in the approach to apply metaheuristics for musical style modification, but is not about learning the objectives from a data corpus in the way described here.

Having mentioned some major branches of automatic music generation, the author recommends a more complete survey [25] for those interested in more branches of this field.

7. CONCLUSIONS

In this paper a novel approach of musical style modification has been presented. Basically this is a multi-objective optimization, where the objectives try to reward similarity to the target style in different respects. By that means a given piece of music can be transformed with the aim of pushing it closer to a specified target style. There are plenty of possibilities to built upon this work: making it real-time capable (currently it is not), paying more regard to the metrical structure (a weakness in the Pastorius-project),

⁸ That fits since in Cope’s terminology Markovian processes fall into this category

validly evaluating the modification results by empirical experiments. We also started to conceptualize about pedagogical applications: during the modification process, *reasons* why things has been changed in certain manner, can be tracked—something that could be expanded for automatically explaining style. This shows that the potential of this approach is far from exhausted.

Acknowledgments

This paper is a condensed form of a Master thesis [4], so thanks are owed to all who supported this thesis: First and foremost SAP SE who generously supported it in financial terms, by a providing a great working environment as well as with personnel support: Stephan Brand who initiated the project and debated about the project from the perspective of a bassist, Pastorius enthusiast and software manager; Benjamin Graf who was available weekly for discussions especially about data science issues. Special thanks to Prof. Denis Lorrain whose encouraging and far-sighted arguments have been inspiring as well as Sean Malone, who contributed by occasional mail contact with beneficial comments from his view as bassist, musicologist and Pastorius expert.

8. REFERENCES

- [1] G. Adler, *Der Stil in der Musik. 1. Buch: Prinzipien und Arten des Musikalischen Stils*. Leipzig: Breitkopf & Härtel, 1911.
- [2] L. B. Meyer, “Meaning in Music and Information theory,” in *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, 1957, pp. 412–424.
- [3] S. Luke, *Essentials of Metaheuristics*, 2nd ed. Raleigh: Lulu, 2013. [Online]. Available: <https://cs.gmu.edu/~sean/book/metaheuristics/>
- [4] F. Zalkow, “Automated musical style analysis. Computational exploration of the bass guitar play of Jaco Pastorius on symbolic level,” Master’s Thesis, University of Music Karlsruhe, Germany, Sep. 2015.
- [5] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The elements of statistical learning. Data mining, inference, and prediction*, ser. Springer series in statistics. New York: Springer, 2009.
- [6] M. S. Cuthbert, C. Ariza, and L. Friedland, “Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit,” in *Proceedings of the 12th International Symposium on Music Information Retrieval*, 2011, pp. 387–392.
- [7] S. Malone, *A Portrait of Jaco. The Solo Collection*. Milwaukee: Hal Leonard, 2002.
- [8] E. Alpaydın, *Introduction to Machine Learning*, 2nd ed., ser. Adaptive computation and machine learning. Cambridge and London: MIT Press, 2010.
- [9] F. Jelinek and R. L. Mercer, “Interpolated estimation of Markov source parameters from sparse data,” in *Proceedings, Workshop on Pattern Recognition in Practice*. Amsterdam: North Holland, 1980, pp. 381–397.
- [10] F. Pachet and P. Roy, “Markov constraints. Steerable generation of Markov sequences,” in *Constraints*, vol. 16, no. 2, March 2011, pp. 148–172.
- [11] S. F. Chen and J. Goodman, in *An empirical study of smoothing techniques for language modeling*, vol. 13, no. 4, 1999, pp. 359–393.
- [12] D. Cope, *Computers and Musical Style*. Madison: A-R Editions, Inc., 1991.
- [13] ———, *Virtual Music. Computer Synthesis of Musical Style*. Cambridge: MIT Press, 2001.
- [14] ———, *Computer Models of Musical Creativity*. Cambridge: MIT Press, 2005.
- [15] F. Brooks, A. Hopkins, P. Neumann, and W. Wright, “An experiment in musical composition,” in *IRE Transactions on Electronic Computers*, vol. 6, no. 3, 1957, pp. 175–182.
- [16] L. Hiller and L. Isaacson, *Experimental Music. Composition With an Electronic Computer*. New York: McGraw-Hill, 1959.
- [17] I. Xenakis, *Formalized Music. Thought and Mathematics in Composition*, 2nd ed., S. Kanach, Ed. Stuyvesant: Pendragon Press, 1992.
- [18] R. C. Pinkerton, “Information theory and melody,” in *Scientific American*, vol. 194, no. 2, 1956, pp. 77–86.
- [19] J. E. Youngblood, “Style as Information,” in *Journal of Music Theory*, vol. 2, no. 1, 1958, pp. 24–35.
- [20] F. Pachet, P. Roy, and G. Barbieri, “Finite-length Markov Processes with Constraints,” in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, vol. 1, 2011, pp. 635–642.
- [21] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, “Using Machine-Learning Methods for Musical Style Modeling,” *Computer*, vol. 36, no. 10, pp. 73–80, Oct. 2003.
- [22] J. Nika and M. Chemillier, “Improtek: integrating harmonic controls into improvisation in the filiation of OMax,” in *Proceedings of the 2012 International Computer Music Conference*, 2012, pp. 180–187.
- [23] A. Donzé, R. Valle, I. Akkaya, S. Libkind, S. A. Seshia, and D. Wessel, “Machine Improvisation with Formal Specifications,” in *Proceedings of the 40th International Computer Music Conference*, 2014, pp. 1277–1284.
- [24] T. Dimitrios and M. Elen, “A GA Tool for Computer Assisted Music Composition,” in *Proceedings of the 2007 International Computer Music Conference*, 2007, pp. 85–88.
- [25] J. D. Fernández and F. Vico, “AI Methods in Algorithmic Composition: A Comprehensive Survey,” in *Journal of Artificial Intelligence Research*, vol. 48, no. 1, 2013, pp. 513–582.