Melody Blending: A Review and an Experiment

Stefano Kalonaris[1] and Omer Gold[2]*

[1] Music Information Intelligence Team, RIKEN AIP, Japan
[2] Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. The blending of two melodies into a third is a creative process useful for exploring a search space and can be employed in compositional or improvisational tasks. Two melodic blend tropes are considered: *hybridization* (recombination of features) and *morphing* (generation of intermediate feature values). After reviewing the approaches that have been used to this end, a bespoke implementation of common methods for both tropes is undertaken, and excerpts demonstrating some use case scenarios are provided. A set of evaluation metrics is then put forward and selected blending modes are tested accordingly in a melodic blending task, for comparison.

1 Introduction

In this paper, the task of obtaining a melody C by blending two melodies A and B is considered. The goal is to produce C so to retain perceptual properties of both input melodies, to different degrees and according to different methods. This procedure relates to conceptual blending [1] whereby two input spaces are integrated into a third by cross-mapping and projection. Conceptual blending has been hailed as a useful tool for creative exploration, and has been used in music with applications relating to harmonization [2] or emotion [3], among others. While there are precedents [4] of conceptual blending applied to melody generation, this paper narrows the scope by inheriting the distinction between *hybridization* and *morphing* originally proposed in [5] and porting it from the raw audio domain to symbolic representation. In hybridization, each attribute of C is inherited by A or B. Thus, each constituent part of C is obtained by recombining the respective parts of the input melodies. In morphing, instead, the resulting melody C is an "in between", intermediate melody which typically maintains the shared properties of melodies A and B (if they exist), and can be closer to A or to B, proportionally to a morphing coefficient λ. Hereinafter, the terms *source*, *target*, and *blendoid* will be used interchangeably with A, B, and C, respectively.

Different approaches (ranging from simple *recombination* [6], to music theory [7], or even number theory [8]) have been used to implement melodic blending, each with its own advantages and limitations. The most notable of these are reviewed in Section 2,

* Part of the work on this paper was done while the second author was visiting RIKEN, Japan.

revealing frequent misnaming and ambiguity (according to the hybridization/morphing dichotomy), as well as much diversity regarding evaluation procedures and metrics. With focus on their creative potential for music generation and computer aided composition, some of the blending methods reviewed are reimplemented ad hoc in Section 3, use case scenarios are illustrated in Section 4, and a set of metrics derived from [5] is applied for the evaluation of blended melodies, in Section 5.

2 Related Work

Following is an overview of some key approaches developed so far in the context of melodic blending.

2.1 Music Theory

Hamanaka et al. [9, 10] proposed melody blending methods based on the *Generative Theory of Tonal Music* [11] (GTTM) whereby, after computing the intersection between the time-span trees[3] for melodies A and B, an intermediate melody is generated by combining segments of the two melodic divisional reductions going from each melody to the intersection. Because of the difficulty in applying the (often ambiguous) GTTM preference rules, this method has suffered from a lack of automatization, and requires human expertise (*i.e.*, manual annotation of GTTM tree structures). This method also assumes that the two reference melodies are in the same key and with an non-empty intersection set. According to [9] the melodies generated using this method satisfy the condition that A & C and B & C are more similar than A & B. The measure of similarity is reportedly calculated as the intersections of notes $A \cap B$ scaled by the reciprocal of $\max(\text{length}_A, \text{length}_B)$ and thus does not account for the interpolation of notes. Furthermore, the literature on the GTTM-based blending method only provides examples where melodies A and B are related to each other. Arguably, said examples are more akin to what in music is known as the "theme and variations" practice, rather than blending of two independent melodies. For these reasons, it is unclear whether GTTM-based melodic blending can be fully classed as a morphing method, falling somewhat in between the two blending categories.

2.2 Probability

The probabilistic approach proposed by Wooller & Brown [12] is also difficult to class (although its authors use the term *morphing*). According to it, the input melodies are subdivided into segments of equal duration (in quarter note length). Starting from a source segment and based on a probability value p (which determines whether to sample from either the source or target) and the *order* of the Markov process (how many steps to look back within the pitch and duration sequences), the algorithm generates the next segment, sampling from the chosen Markov chain. This repeats as many times as

[3] one of the four hierarchical structures used in the GTTM, the other three being: grouping structure, metrical structure and prolongational reduction.

desired. This method dissociates "musical segments with their original temporal location" [12] and ignores concerns about the alignment between the source and the target. Nevertheless, it is suitable as a creative tool for generating melodic blends and transformations. Wooller & Brown's method was evaluated through the responses and commentaries of eleven volunteers who compared transitions (both short and long) between tracks performed by a DJ with those obtained by the Markov-based blending. While the focus was on qualitative metrics and the perceived musicality of the blending transitions, the results of this evaluation are difficult to generalize, given the size of the study.

2.3 Geometry

DMorph [13] is Oppenheim's proprietary system which allows the blending of two or more melodies based on Dynamic Time Warping [14] or time syncing algorithms. DMorph affords different methods but, while Oppenheim defines morphing as "the sensation of a natural transformation from one theme into another" [13, p.5], some of these (*e.g.*, recombination, interleaving, weighted selection) might class as hybridization, while others (*i.e.*, interpolation) abide by the formal definition of morphing found in [5]. DMorph is suitable for pairing sections of the source to sections of the target beyond arbitrary length sampling. It is a fully automatic method and does not depend on corpora or domain expert knowledge. Unfortunately, DMorph is not open source and a working version of the software is nowhere to be found. To the authors' knowledge, DMorph lacks a formal evaluation.

2.4 Neural Nets

MusicVAE [15] is a variational autoencoder model which addresses long-term structure by using the embeddings of the input musical subsequences to generate output subsequences independently. To train and generate accordingly, MusicVAE requires monophonic melodies or drum patterns of a specified length. The quantization is done in sixteenth notes based on the assumption that all training points are in a 4/4 meter. For the evaluation of MusicVAE, both quantitative and qualitative methods were used. The former included assessing the accuracy of the MusicVAE in reconstructing melodies and comparing the interpolations of two types of MusicVAE against a baseline obtained by weighted selection. The latter, instead, asked participants to indicate on a Likert scale whether they deemed the model's or real compositions more musical.

A more recent work [16] uses VAE to connect smoothly two musical sequences, where *smoothness* relates to pitch and duration transition (*i.e.*, a few consecutive notes around the connection boundary are used to compute Markov transition matrices of each statistics as states).

3 Melody Blending

Some of the techniques discussed so far are here reimplemented with in view to, in the future, developing an integrated toolbox for melody blending. This section describes the main technical details, to this end.

3.1 Alignment

To blend two melodies, an appropriate alignment between them must be established first. Here, priority for the alignment is given to the time dimension, and two approaches are explored: *time-sync* and *time-warp*.

Time-sync In the authors implementation of time-sync alignment, it is assumed that source and target are reasonably similar in quarter length duration. If needed, the melodies are *zero-padded* (by lengthening the shortest melody with a rest of duration equal to the sequence difference) so that their length match and divide by a quarter note duration. Then, the melodies are partitioned into disjoint segments as follows. The first segment starts at the beginning of each melody and ends once an event (note/pause) in which the cumulative sum of the onset values of the source and target melodies equals to t times quarter note duration is reached, where t is an integer (it is assumed that such a t always exists after the zero-padding). The next segment starts after the end of the previous segment, and so on, until the end of the melodies. An example of this procedure is given in Figure 1.

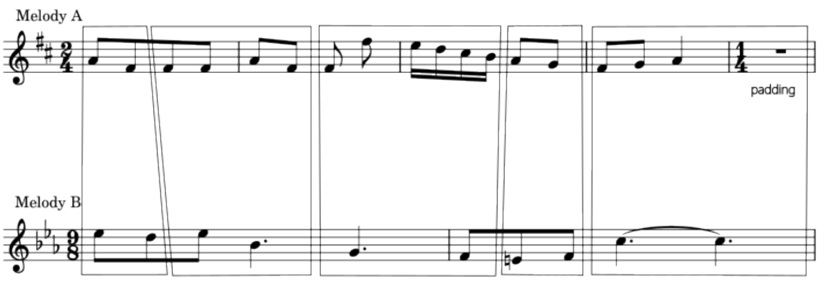

Fig. 1. An example of time-sync alignment between melodic extracts from L.v. Beethoven's *6 Variations in D major, op.76* (melody A, top) and S. Foster's *Beautiful Dreamer* (melody B, bottom), using quarter-note syncing.

Time-warp Dynamic time warping (DTW) [14], instead, is a geometrical approach which can be used also when the source and target melodies are considerably different in length. Recall the definition of DTW between two point-sequences. Let $A = (p_1, \dots, p_n)$ and $B = (q_1, \dots, q_m)$ be two sequences of points in some metric space (X, dist). A DTW-coupling $C = (c_1, \dots, c_k)$ between A and B is an ordered sequence of distinct pairs of points from $A \times B$, such that $c_1 = (p_1, q_1)$, $c_k = (p_n, q_m)$, and $c_r = (p_i, q_j) \Rightarrow c_{r+1} \in \{(p_{i+1}, q_j), (p_i, q_{j+1}), (p_{i+1}, q_{j+1})\}$, for $r < k$ (note that $\max\{n, m\} \leq k \leq n + m$). The DTW-distance between A and B is

$$\mathsf{dtw}(A, B) = \min_{C:\, coupling} \Big\{ \sum_{(p_i, q_j) \in C} \mathsf{dist}(p_i,\, q_j) \Big\}. \tag{1}$$

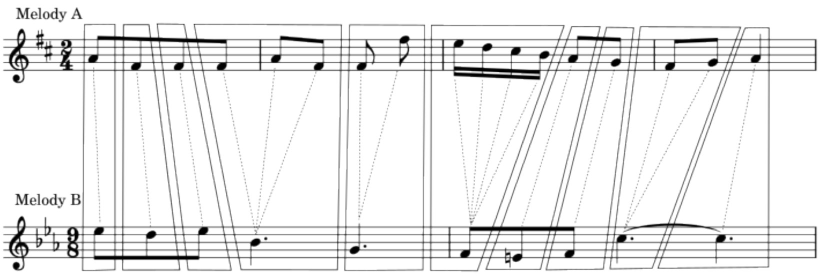

Fig. 2. Example of a time-warp alignment between the onset series of the same melodies given in Figure 1.

A coupling C for which the above sum is minimized is called an *optimal coupling*[4]. Here, the two point-sequences A and B represent melodies, where each point $a_i \in A$ and each point $b_i \in B$ is a vector with entries corresponding to musical features (*e.g.*, pitch, duration, velocity, etc.). The distance metric dist can be chosen among common measures. In this case, the Euclidean distance was used.

Using different feature vectors for the calculation of an optimal coupling (in the fashion of Conklin's *viewpoint sequences* [17]) might produce different results. However, in an effort to achieve better rhythmic coupling, onset series were used as the new point-sequences, to this end. The typical optimal coupling format is a sequence of tuples of indices for matching events in A and B. For example, the optimal coupling in Figure 2 would be:

$$C = (0,0),(1,1),(2,2),(3,3),(4,3),(5,3),(6,4),(7,4),(8,5),(9,5),\ldots \quad (2)$$

3.2 Blend Methods

Based on the precedents seen in Section 2, several methods for melodic transformation were implemented or adapted.

Interleaving In this blend method, one simply alternates between source and target, using the matched events obtained either by time-sync or time-warp alignment (as specified by the user). In the example used thus far, matched events are clearly delineated using polygon contours (see Figures 1 and 2). Despite its simplicity, the interleaving method can produce some interesting blends (see Section 4 for an example).

Weighted Selection This blend method operates similarly to interleaving, but considers a blend coefficient between 0 and 1 as the probability p of selecting, for a given match, from either the source or the target. Weighted selection affords the ability to steer the output closer to the source or the target.

[4] It is possible that there is more than one optimal coupling.

Markov Chain Similarly to the previous method, for each match, either the source or the target is selected stochastically using the blend coefficient. Accordingly, the first event is used as the seed to generate a sequence of notes/rests based on the corresponding transition matrix (source or target), using a specified Markov order, and for as long as the duration sum of the generated events (in quarter length) does not exceed that of the original events in the match. As an example, consider the subsequence $(3,3),(4,3),(5,3)$ of the optimal coupling (2). Suppose that according to the blending coefficient the source is selected: then, the event with index 3 in the source (*i*.e., an *F#* eight note) will be the seed for generating notes/rests based on the source's transition matrix, for as long as their duration sum does not exceed a dotted quarter note, which is the duration sum for events with indices $(3,4,5)$ in the source. This process repeats until the exhaustion of matched events in the alignment.

Interpolation This blend procedure uses pitch and duration value interpolation over a time-warp optimal coupling. Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_m)$ be two melodies, where each point $a_i \in A$ and $b_j \in B$ is a vector with entries corresponding to musical features (*e.g.*, pitch, duration, velocity, etc.). Let $C = (c_1, \ldots, c_k)$ be an optimal coupling obtained by the DTW algorithm. For each pair $c = (a_i, b_j) \in C$, the musical features are interpolated so that, for each pair of points in the coupling, a new point that is "in-between" them is obtained. Although there are many interpolation techniques (piecewise constant, spline, etc.), in the authors' system, the morphed feature $m_{i,j}$ for a pair $c = (a_i, b_j) \in C$ is generated by applying linear interpolation using a blend coefficient to yield intermediate values closer to either the source or the target, as desired.

4 Use Cases

Different blending methods may be more or less appropriate depending on the musical task at hand.

4.1 Style Blend

For example, if one wanted to blend styles in a given musical genre, pure interpolation methods could prove problematic for idiomatic dependencies that might be expected in a scenario of this kind. Conversely, weighted selection or Markov-based methods might be better candidates. Figure 3 shows II-V-I[5] *licks*[6] by C. Parker's solo on *Au Privave* and from M. Brecker's solo on *Take a Walk*, and the blendoid obtained using weighted selection with a 0.3 blend coefficient.

Source and target are indicative of how the jazz idiom developed over the years, from the *enclosure* approach [18] common in the *be bop* era to the polychordal superimposition employed by more recent players, and the blendoid is an example of successful hybridization of the two.

[5] A standard chord progression serving as building block for larger harmonic structures.
[6] Idiomatic melodic patterns.

Fig. 3. Blending styles over a II-V-I chord progression using weighted selection with time-sync and a blend coefficient of 0.3.

4.2 Theme & Variations

Another task where time-sync is suitable could be the generation of variations, as commonly done in the classical tradition. Figure 4 shows a possible variation in the context of W. A. Mozart's *7 Variations on "Willem von Nassau"*, K.25, obtained by blending the original theme with the 3^{rd} variation.

Fig. 4. A blendoid (bottom stave) generated by interleaving the theme (top stave) and the 3^{rd} variation (middle stave) of *7 Variations on "Willem von Nassau"*, K.25 by W.A. Mozart.

4.3 Heterogeneous Blend

A case where time-warp interpolation methods would prove interesting is the blending of melodies from heterogeneous genres, or with different metrical structures, length, etc. As an example, Figure 5 shows an interpolation blend of *Le Cygne* by C. Saint-Saëns and *Salut d'amour* by E. Elgar, using a 0.3 coefficient.

Fig. 5. A blendoid (bottom block) generated by interpolating ($\lambda = 0.3$) excerpts of *Le Cygne* by C. Saint-Saëns (top block) and *Salut d'amour* by E. Elgar (middle block).

5 Evaluation

As seen in Section 2, there is no standardized procedure for evaluating melodic blends. Given the combination of available blending methods and time alignments, a universal and exhaustive evaluation protocol might be beyond the scope of this paper. In fact, important criteria in the evaluation of morphing methods might not have a clear correspondence for hybridization techniques and viceversa, thus making the development of consistent evaluation metrics difficult. Notwithstanding, and deferring a more comprehensive evaluation framework to include qualitative metrics to future endeavors, a minimal set of objective metrics is tested. These include *similarity* and two of the three independent criteria proposed in [5]: *intermediateness* and *smoothness*. It must be noted that the latter were originally developed for raw audio and are here interpreted and implemented to reflect the different representation (symbolic) of the musical surface. *Correspondence*, originally also part of the set in [5], is not contemplated here, as one assumes it is guaranteed by virtue of the feature matching in the representation of the melodies. Only blending methods allowing a blend coefficient were considered in this study: weighted selection, Markov chain, and interpolation. These are evaluated over complete blends, going from 0.0 to 1.0 with 0.1 increments, as described below.

Similarity Many melodic similarity measures have been proposed and argued, the main approaches being mathematical [19–29], cognition-based [30–32], and musicological [33–35]. To account for true in-between pitch values, this study focuses on melodic contours and employs two measures. One is obtained as in [35], albeit substituting the original *n-gram* similarity over the extended Implication-Realization (IR) symbols at character level with the complement of the *n-gram Jaccard* similarity at token level. The other similarity measure is obtained using the normalized Euclidean DTW distance between melodic contour (smoothed) series. For either of these similarity measures $\mathrm{sim}(\cdot,\cdot)$, the indicator function in Equation 3 determines whether a blendoid b is appropriately more similar to the source s or the target t with respect to the blend coefficient λ. The weighted sum over a complete blend is taken as the final measure and indicated as SimIR or SimDTW, depending on which similarity metric was used for the indicator function.

$$\mathbf{I}(s,t,b) := \begin{cases} 1 & \text{if } (1-\lambda)\cdot\mathrm{sim(s,b)} \geq \lambda\cdot\mathrm{sim(t,b)}, \text{for } \lambda \leq 0.5 \\ 0 & \text{if } (1-\lambda)\cdot\mathrm{sim(s,b)} < \lambda\cdot\mathrm{sim(t,b)}, \text{for } \lambda \leq 0.5 \\ 1 & \text{if } \lambda\cdot\mathrm{sim(t,b)} \geq (1-\lambda)\cdot\mathrm{sim(s,b)}, \text{for } \lambda > 0.5 \\ 0 & \text{if } \lambda\cdot\mathrm{sim(t,b)} < (1-\lambda)\cdot\mathrm{sim(s,b)}, \text{for } \lambda > 0.5 \end{cases} \tag{3}$$

Intermediateness For intermediateness, a problem posed by the symbolic music domain is the limited choice of discrete steps for in-between notes. Another issue to bear in mind is that linear interpolation of the parametric space does not necessarily result in perceptually intermediate blends. Notwithstanding, the following procedure is proposed: first, the melodic piecewise contours for source s, target t, and blendoid b, are calculated and resampled to n points proportionally to the blending coefficient. Then, for each point i in this range, the following is checked: $min(s_i,t_i) \leq b_i \leq max(s_i,t_i)$. The weighted sum of all the `True` values is taken as the intermediateness index for that blendoid.

Smoothness In [36], a melody is defined smooth simply if the intervals between consecutive notes are within a fifth (*i.e.*, seven semitones). In the context of this experiment, however, a different definition is needed to compare melodies and to quantify whether the blending from source to target is gradual and, thus, successful. In this paper, *autocorrelation* (lag-one), *roughness*, and *mean squared jerk* (MSJ) are employed. Autocorrelation with scores near 1 might imply a smoothly varying series whereas if there isn't an overall linear relationship between consecutive data points one might expect values closer to 0. Roughness in this context is considered as the smoothness penalty as defined in the cubic spline, albeit with a normalization factor that accounts for the length of the input series. The mean squared jerk measure is defined as in [37], and here adapted to the music domain (it is normally employed in movement analysis to measure how much the acceleration of a movement contour changes over time). For all three smoothness measures, the melodic contour (smoothed) series of each blendoid in a complete blend is used as input (like in the DTW-based similarity described earlier).

Using the above metrics and the same two melodic excerpts of Section 3.1, yielded the results shown in Table 1. Note that the values (mean and standard deviation) reported refer to a run of 10 instances of complete blends since all methods but interpolation are stochastic and might generate different blendoids for the same blend coefficient. For the Markov-based method, an order of $n = 3$ was used.

Table 1. Comparing different blending methods based on the proposed evaluation metrics, over 10 full blends. Abbreviations for the methods are: WS (weighted selection), MC (Markov chain), and Lerp (linear interpolation), with *ts* and *tw* indicating time-sync and time-warp, respectively. Abbreviations for the evaluation metrics are: Intrm (intermediateness), Acorr (autocorrelation), Rghns (roughness), and MSJ (mean squared jerk).

	SimIR	SimDTW	Intrm	Acorr	Rghns	MSJ
WS (ts)	0.9 $\pm$ 0.3	**0.833** $\pm$ 0.373	0.405 $\pm$ 0.045	0.988 $\pm$ 0.005	2.185 $\pm$ 1.412	2.889 $\pm$ 2.045
WS (tw)	0.878 $\pm$ 0.328	0.722 $\pm$ 0.448	**0.43** $\pm$ 0.13	0.986 $\pm$ 0.007	3.95 $\pm$ 3.006	4.118 $\pm$ 2.912
MC (ts)	**0.911** $\pm$ 0.285	0.689 $\pm$ 0.463	0.372 $\pm$ 0.069	0.99 $\pm$ 0.004	2.3 $\pm$ 1.943	3.071 $\pm$ 3.054
MC (tw)	0.878 $\pm$ 0.328	0.822 $\pm$ 0.382	0.359 $\pm$ 0.06	0.99 $\pm$ 0.002	2.295 $\pm$ 1.528	2.654 $\pm$ 2.189
Lerp	0.778 $\pm$ 0.416	0.444 $\pm$ 0.497	0.402 $\pm$ 0.041	**0.994** $\pm$ 0.001	**0.527** $\pm$ 0.352	**0.507** $\pm$ 0.413

6 Conclusion

This paper offered a brief review of melodic blending approaches, presented an original appropriation for some of these, and proposed objective metrics, in an effort to move towards a more standardized evaluation procedure. The blending operations implemented by the authors are prototypical, and much remains to be improved upon. The morphing methods, particularly, do not handle diatonic perceptual imperatives, and, in cases with a strong "tonal" or "idiomatic" expectation, linear interpolation of features is likely to violate it. Additional features (*e.g.*, dynamics, articulation), could also be included to enhance the blended melody's musical quality. It is also important to note that, while this experiment dealt with standard symbolic representation, there are other approaches, such as the Tonal Interval Space [38], which merit consideration in future implementations, as they might yield different and more nuanced intermediate values for interpolation. Despite the system's current limitations, this experiment's results suggest that a toolbox packaging of the blending functionality described in this paper could be a useful addition to one's creative workflow, either as a module in a larger generative music system or, conditioned upon further development, as a standalone application.

References

1. Gilles Fauconnier and Mark Turner. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books, New York, NY, 2003.
2. Maximos Kaliakatsos-Papakostas, Marcelo Queiroz, Costas Tsougras, and Emilios Cambouropoulos. Conceptual blending of harmonic spaces for creative melodic harmonisation. *Journal of New Music Research*, 46(4):305–328, 2017.
3. Michael Spitzer. Conceptual blending and musical emotion. *Musicae Scientiae*, 22(1):24–37, 2018.

4. Maximos Kaliakatsos-Papakostas. Generative conceptual blending of high-level melodic features: Shortcomings and possible improvements. In *International Conference on New Music Concepts (ICNMC)*, 2019.
5. Marcelo Caetano and Naotoshi Osaka. A formal evaluation framework for sound morphing. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 104–107, 2012.
6. David Cope. *Computer Models of Musical Creativity*. The MIT Press, Cambridge, MA, 2005.
7. Masatoshi Hamanaka and Keiji Hirata anSatoshi Tojo. Applying melody morphing method to composition. In *Proceedings of the 3rd Conference on Computer Simulation of Musical Creativity (CSMC)*, 2018.
8. Jay Hardesty. A self-similar map of rhythmic components. *Journal of Mathematics and Music*, 10(1):36–58, 2016.
9. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Melody Morphing Method Based on GTTM. In *Proceedings of the International Computer Music Conference (ICMC)*, 2008.
10. Keiji Hirata, Satoshi Tojo, and Masatoshi Hamanaka. Melodic Morphing Algorithm in Formalism. In Carlos Agon, Moreno Andreatta, Gérard Assayag, Emmanuel Amiot, Jean Bresson, and John Mandereau, editors, *Mathematics and Computation in Music*, pages 338–341, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
11. Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA, 1983.
12. René Wooller and Andrew R. Brown. Note sequence morphing algorithms for performance of electronic dance music. *Digital Creativity*, 22:13–25, 2011.
13. Daniel Oppenheim. 'DMorph': An Interactive System for Compositional Morphing of Music in Real-Time. Technical report, IBM, 1995.
14. Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4), 2018.
15. Adam Roberts, Jesse H. Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *CoRR*, abs/1803.05428, 2018.
16. Taketo Akama. Connective fusion: Learning transformational joining of sequences with application to melody creation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pages 46–53, Montreal, Canada, October 2020.
17. Darrell Conklin and Christina Anagnostopoulou. Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 479–485, 2001.
18. Jerry Coker. *Elements of the jazz language for the developing improvisor*. Alfred Publishing Company, 1991.
19. Luigi Logrippo and Bernard Stepien. Cluster analysis for the computer-assisted statistical analysis of melodies. *Computers and the Humanities*, 20(1):19–33, 1986.
20. Donncha Ó Maidín. A geometrical algorithm for melodic difference. In Walter B. Hewlett and Eleanor Selfridge-Field, editors, *Melodic Similarity - Concepts,Procedures and Applications, Computing in Musicology II*, chapter 2, pages 65–72. MIT Press, Cambridge, Massachusetts, 1998.
21. Rodger J. McNab, Lloyd A. Smith, Ian H. Witten, Clare L. Henderson, and Sally Jo Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the First ACM International Conference on Digital Libraries*, DL '96, pages 11–18, New York, NY, USA, 1996. ACM.
22. Kjell Lemström. *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Faculty of Science, Department of Computer Science, 2000.

23. Godfried Toussaint. A comparison of rhythmic similarity measures. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 242–245, 2004.
24. Rainer Typke, Panos Giannopoulos, Remco C. Veltkamp, Frans Wiering, and Reneé van Oostrum. Using transportation distances for measuring melodic similarity. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 107–114, 2003.
25. Shyamala Doraisamy and Stefan Rüger. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70, 2003.
26. Anna Lubiw and Luke Tanur. Pattern matching in polyphonic music as a weighted geometric translation problem. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004.
27. Rainer Typke, Frans Wiering, and Remco C. Veltkamp. Transportation distances and human perception of melodic similarity. *Musicae Scientiae*, 11(1_suppl):153–181, 2007.
28. Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. Melodic similarity through shape similarity. In Sølvi Ystad, Mitsuko Aramaki, Richard Kronland-Martinet, and Kristoffer Jensen, editors, *Exploring Music Contents*, pages 338–355, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
29. Fumio Hazama. Mathematical analysis of melodies: Slope and discrete Frechet distance. *ArXiv e-prints*, 2014.
30. Ana de Carvalho Junior and Louis Batista. Sms identification using PPM, psychophysiological concepts, and melodic and rhythmic elements. In *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 2012.
31. Carles Roig, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Submission to Mirex 2013 symbolic melodic similarity. In *Proceedings of the Annual Music Information Retrieval Evaluation Exchange*, 2013.
32. Naresh N. Vempala and Frank A. Russo. An empirically derived measure of melodic similarity. *Journal of New Music Research*, 44(4):391–404, 2015.
33. Maarten Grachten, Josep Lluís Arcos, and Ramón López de Mántaras. Melody retrieval using the implication / realization model. In *Proceedings of the Annual Music Information Retrieval Evaluation Exchange*, 2005.
34. Nicola Orio and Antonio Rodà. A measure of melodic similarity based on a graph representation of the music structure. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2009.
35. Sakurako Yazawa, Masatoshi Hamanaka, and Takehito Utsuro. Subjective melodic similarity based on extended implication-realization model. *International Journal of Affective Engineering*, 15(3):249–257, 2016.
36. Rui Pedro, Paiva Teresa, and Mendes Amílcar Cardoso. Exploiting melodic smoothness for melody detection in polyphonic audio. In *Proceedings of the International Computer Music Conference (ICMC)*, 2005.
37. Neville Hogan and Dagmar Sternad. On rhythmic and discrete movements: reflections, definitions and implications for motor control. *Experimental Brain Research*, 181(1):13–30, Jul 2007.
38. Gilberto Bernardes, Diogo Cocharro, Marcelo Caetano, Carlos Guedes, and Matthew E.P. Davies. A multi-level tonal interval space for modelling pitch relatedness and musical consonance. *Journal of New Music Research*, 45(4):281–294, 2016.