

MUSICMEAN: FUSION-BASED MUSIC GENERATION

Tatsunori Hirai

Waseda University

tatsunori_hirai@asagi.waseda.jp

Shoto Sasaki

Waseda University

/ CREST, JST

Shigeo Morishima

Waseda Research Institute
for Science and Engineering

/ CREST, JST

shigeo@waseda.jp

ABSTRACT

In this paper, we propose *MusicMean*, a system that fuses existing songs to create an “in-between song” such as an “average song,” by calculating the average acoustic pitch of musical notes and the occurrence frequency of drum elements from multiple MIDI songs. We generate an in-between song for generative music by defining rules based on simple music theory. The system realizes the interactive generation of in-between songs. This represents new interaction between human and digital content. Using *MusicMean*, users can create personalized songs by fusing their favorite songs.

1. INTRODUCTION

Because composing songs from scratch is difficult for inexperienced people, musical composition used to be the act of expression only approved for people with musical sense. However, the development of music creation software has made it easy for a wide range of people to create music. Despite easy-to-use music creation software, some people, particularly those who are not accustomed to expressing their feelings through music, have difficulty creating original music. Therefore, we propose *MusicMean*— a music creation system that allows users to create songs by fusing existing songs.

Music evokes subjective impressions. For example, people feel that “this song is good” or “this song lacks meaning.” Such impressions are spontaneous and occur even though the listener does not have a good grounding in music theory or does not possess musical sense. These impressions can be a source of creativity. *MusicMean* enables users to create a new song interactively by fusing songs depending on the user’s impressions of existing songs. Thus, people with little musical knowledge or experience can construct personalized music content while listening to the fused music. We call this fused music an “in-between song.” While this may not actually be original song writing, it appeals to an essential motivation for music composition. *MusicMean* represents a small step toward an era in which everyone can create personalized musical content. Our goal is to realize such a content creation environment.

We also propose the concept of an “average song,” as a part of in-between song, by calculating the average¹ of musical elements, e.g., notes from several songs.

In most circumstances, listeners cannot change an existing song. Listeners had to listen to the song as it is unless they arrange it. Music has always been what listeners listen to, but what listeners do not create. Our objective is to realize a new interaction between listeners and musical content. *MusicMean* allows listeners to alter a song to suit their preference via exploration of space in-between existing songs.

In the context of researches about musical experiences for novices, many new instruments, interfaces or applications are proposed. Blaine and Fels explored the context of the research on collaborative musical experiences for novices including music composition experiment and introduced them on [1]. *MusicMean* also provides user a new musical experience using existing music rather than creating whole new sound from scratch. To support novice users compose music, there are researches area called Computer-Assisted Composition (CAC) [2–5]. *MusicMean* can also be regarded as an application in the context of CAC with a possibility to expand creativity of novice users. At the same time, *MusicMean* is music exploration tool so that the user does not have to intend to compose but enjoy the generated song itself.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. We present an outline of the proposed system in Section 3. The averaging concept and its related calculations are discussed in Section 4. Results, conclusions, and suggestions for future work are given in Section 5.

2. RELATED WORK

Research has been performed on the creation of new songs or components of a song, e.g., melody, using existing songs. Melody morphing is a representative technique to fuse two melodies. Hamanaka *et al.* [6] proposed a method to morph two melodies using generative theory of tonal music (GTTM) [7]. The GTTM makes it possible to morph melodies based on notes common to two melodies. However, requiring common notes makes it difficult to generate a fused melody from any two input melodies. In addition, the melody morphing is the method to naturally transform one melody to another melody seamlessly which does not

¹ In this paper, the term “average” refers to the equally mix and the term “in-between” refers to the arbitrary mix.

focus on the intermediate melody. Therefore, our proposed method constructs songs using an averaging operation to generate in-between notes rather than considering generative music models which is used in a melody morphing method. Melody morphing can generate a new melody by fusion; however, our goal is to generate a completely new song. Therefore, we consider fusing rhythm components, e.g., drum sequences and melody.

Wooller and Brown described a music morphing method in their survey paper [8]. The approaches introduced in that paper describe a morphing method for the essence of music rather than morphing the entire song. To the best of our knowledge, the only system that actually morphs songs is *MMorph*, which was proposed by Oppenheim [9]. *MMorph* provides a music morphing interface with an input of up to four songs and several user input morphing algorithms for each music element. Although the user input can help people fuse music, it may complicate the fusion experience. Therefore, we realize the fusion of songs without the selection of detailed parameters. The proposed system requires only the selection of songs and a mixing rate; thus, users are free to fuse songs intuitively.

Reusing existing content to make new music content is a style of song fusion. For example, Hoffman *et al.* [10] turned a Young MC song into an MC Hammer song using spectral matching with Markov chain Monte Carlo sampling. This approach of taking databases of recorded sounds and attempt to combine them to produce a sound matching a target specification is called audio mosaicing. Hoffman’s method is a probabilistic approach that realizes audio mosaicing. In audio mosaicing approach, the audio signals of songs are considered; however, melodies or symbolic notes are not considered. To generate a new song, more semantic and symbolic information such as melody should be considered. To handle melody, we use the MIDI file format as an input music sample.

Reusing existing content to make new content is an established approach in content creation. Some research has generated new music videos by mixing existing music video content. Hirai *et al.* [11] proposed a mashup music video generation system that reuses video content based on audio-visual synchronization. Nakano *et al.* [12] proposed *DanceReProducer*, which is a mashup music video authoring system that employs a statistical audio-visual model. Note that the mashup of content is suitable for beginners, and Davies *et al.* [13] proposed *AutoMashUpper*, a system to mashup multiple songs according to a mashability measure. However, such mashup approaches reuse original content; therefore, the resulting content is a mixture of the original content rather than new and fused content. We aim to make new content which inherit the mood of original songs. In this context, the difference between new and original may seem ambiguous. However, through fusion, the songs MusicMean can generate will bridge the boundary of new and original songs.

Music generation methods based on statistical models have also been proposed [14, 15]. This approach generates new song based on a generative model constructed from training data (i.e., songs). However, such systems are lim-

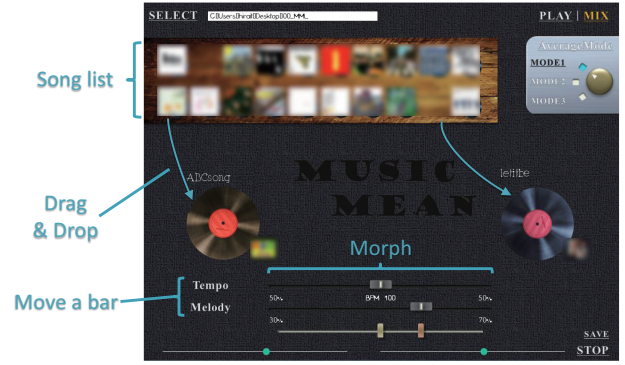


Figure 1. MusicMean screen capture.

ited in how they can reflect user preference because such statistical models cannot be constructed intuitively. Taking these factors into account, our goal is to realize a system that can fuse existing songs intuitively with minimal user input. The proposed system uses a mathematical averaging operation for the fusion of songs rather than the construction of a model for content generation.

3. PROPOSED SYSTEM

Fig. 1 is a screen capture of the proposed MusicMean system. In MusicMean, the MIDI files of each song and a blend rate, which represents a ratio of fusion, are the inputs. The system calculates in-between musical notes using the notes from the source song and the blend rate. Once the user has selected a song from a list, the system plays it. After adding more song and providing a blend rate, the proposed system calculates the in-between musical notes and plays the resulting in-between song in real time. The user can listen to the in-between song and interactively change the blend rate until the system plays a song that satisfies the user. The user can create an in-between song by simply dragging and dropping the songs they wish to fuse and setting the blend rate with a sliding bar. There are two blending parameters that the user can tune, i.e., tempo and melody. By moving a sliding bar, the user can morph songs from one to the other in an aspect of tempo and melody respectively.

MusicMean can also fuse more than two songs. Fig. 2 shows a screen capture of the multisong mixture mode. Each song is allocated to a vertex of an n-sided polygon, and the user can mix songs by moving a control point. The multisong mixture mode allows users to generate an average song for a specific musician or album.

4. FUSION METHOD

We use MIDI files as input to MusicMean. As a result, we do not have to consider sound source separation. MusicMean fuses melody and rhythm parts separately by using an averaging operation to generate an in-between song. The system calculates the in-between musical notes from the input MIDI files and the user-specified blend rate.

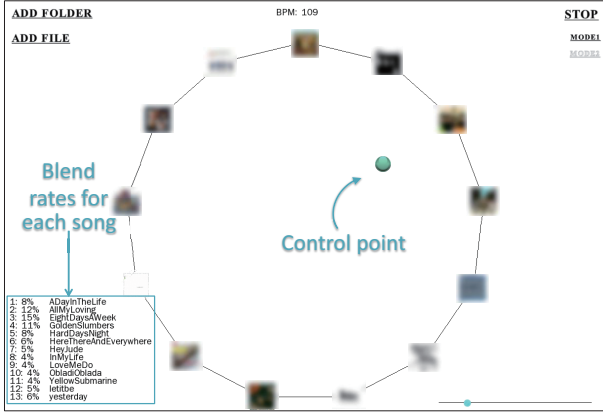


Figure 2. Generating an average song with more than two songs (Making an average song of a specific artist.)

4.1 Scope of consideration in MusicMean

Because we handle MIDI files as input, the output sound is also in MIDI format. Therefore, we cannot consider some elements such as timbre of the songs. Although the instrument type can be considered by a MIDI parameter, we are planning to handle these factors in our future research and only focus on the song itself for the current version.

Therefore, MusicMean does not handle singing. In addition, the number of instruments between fusing songs should be same in current implementation. Accordingly, we define the “song” as the music composed of one or more melody tracks (instruments) and a rhythm track.

4.2 Musical note averaging operation

When the user-specified blend rate is 0.5, the system generates average musical notes based on a geometric mean operation. An average note can be calculated using the pitch f_1 of a note of one song and pitch f_2 of a note of another song. The frequency (pitch) \bar{f} of the average note is calculated using the following equation.

$$\bar{f} = \sqrt{f_1 \times f_2} \quad (1)$$

For example, for C4 (261.2 Hz) and E4 (329.6 Hz) notes with a blend rate of 0.5 (50%), the average note will be 293.7 Hz, which is the pitch of D4. Here, the average frequency may be a sound that is not associated with a musical note. In this case, the note will be rounded off to the nearest musical note in 12 equal temperament. Fig. 3 illustrates the averaging of musical notes.

This averaging operation can be extended to generating an in-between note. An in-between note can be calculated using pitch f_1 of one song’s note and pitch f_2 of another song’s note. With the blend rate α , the frequency (pitch) f' of the average note is calculated using the following equation.

$$f' = f_1^\alpha \times f_2^{1-\alpha} \quad (2)$$

When the blend rate α is 0.5, Eq. (2) corresponds to the geometric mean. Here, the output frequency value will be rounded off to obtain a musical note.

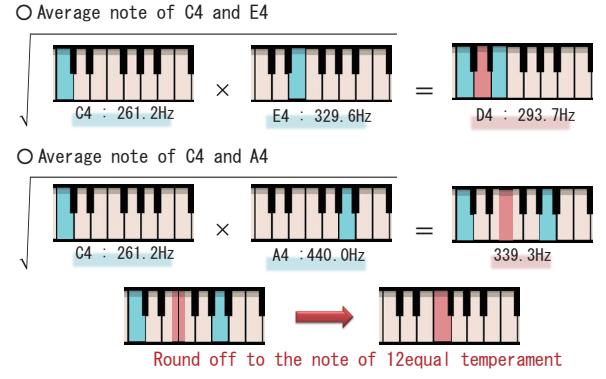


Figure 3. Musical note averaging operation.

4.3 Melody averaging operation

To generate an in-between song, we must consider melodies. Fig. 4 shows the flow for handling melodies. To apply the averaging operation, the system first decomposes all notes of each song into sixteenth notes to align the length of all notes. The proposed system then compares each sixteenth note from the beginning of the both songs. After the averaging operation is performed, the system recombines all the sixteenth notes such that the length of each note is as close as possible to the shortest note among the original notes.

4.4 Generating musical sound

By applying only the averaging operation, the obtained in-between melody will be a series of notes that may not seem to be arranged as a piece of music. MusicMean makes the resulting sound more musical by considering basic music theory. Before calculating an in-between frequency, the system estimates the musical key of the in-between song. By acquiring a histogram of each note from a song, a chromagram for each song can be generated. The weighted sum of the chromagram corresponds to the chromagram of the in-between song. Here, the weight is the blend rate and the musical key of an in-between song is determined based on the top seven notes of the chromagram.

Using the generated musical key, the system estimates chords in each musical bar with reference to music theory. Thus, a musical key of whole song and chords for each bar of music can be determined. Finally, the system rounds off the in-between frequency to the pitch of nearest musical note that composes the chord in each musical bar. Thus, all musical notes in an in-between song become a note from 12 equal temperament and the melody of an average song adheres to the basics of music theory which follow the chord.

4.5 Drum averaging operation

The system also calculates average drum patterns by generating a mixture distribution of probabilities for each type of drum in each song. Drum sounds do not represent a specific musical scale (there are exceptional cases such as the vibraphone or the tom-toms). Therefore, the averaging operation for drum parts differs from the above-mentioned

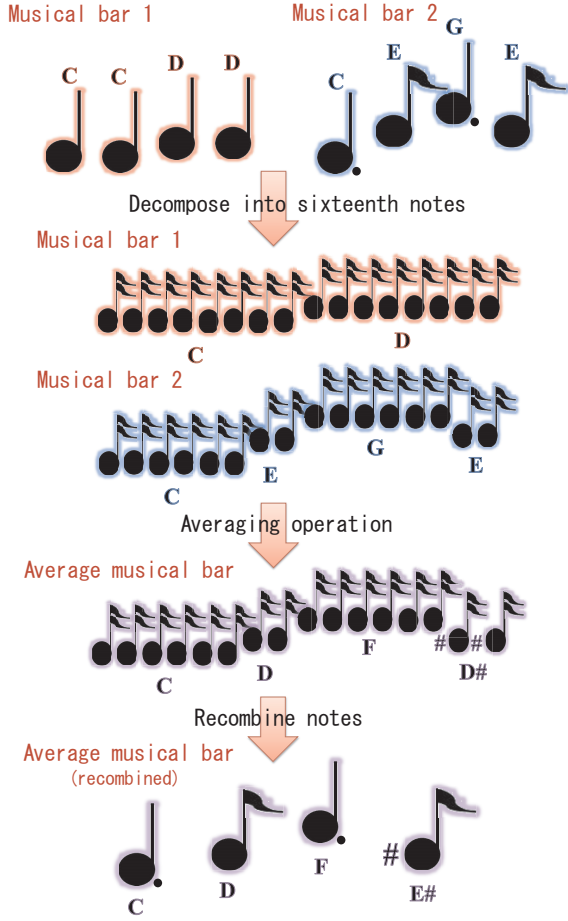


Figure 4. Melody averaging operation.

method. We tested several approaches for averaging drum patterns. We selected a method in which the overall drum pattern does not differ significantly in each musical bar but only changes slightly at regular intervals.

Fig. 5 illustrates the averaging of drum patterns. We describe the drum pattern as a binary sequence of sixteenth beats (0 refers to silence; 1 refers to a sound). For each type of drum, the system generates a drum pattern histogram. The drum pattern histogram describes the time at which a drum sound is produced in a given musical bar. Note that the number of histogram bins is 16. By calculating the weighted sum of the histograms, an in-between drum pattern histogram can be obtained. This histogram indicates the probability of when a drum sound will be produced in a musical bar. If the value is 0.5 for a given timing, the drum will sound at that timing at a 50% probability.

Using this drum pattern histogram, the overall drum pattern will be similar; however, it will differ slightly in each musical bar. In the current implementation of MusicMean, the system does not consider differences in time signature.

Thus, the proposed system outputs an in-between song by calculating the in-between note for each instrument and the in-between drum pattern for each sixteenth note step. This process of generating an in-between song is relatively simple and can run in real time, which allows the user to create and listen to new fused songs interactively.

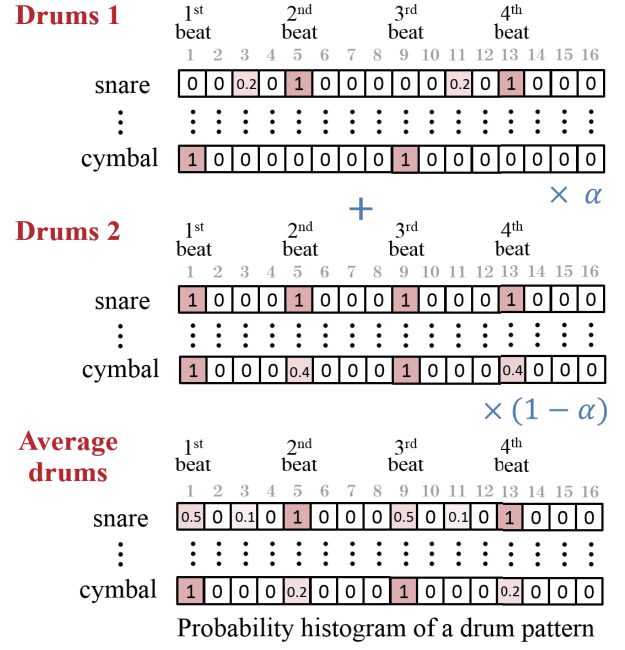


Figure 5. Drum pattern averaging operation.

4.6 “In-betweening” of more than two songs

An in-between song generated from more than two songs can also be obtained as an extension of the above processes. For melody fusing, the in-between frequency (pitch) X of more than two songs can be calculated with the following equation.

$$X = a^\alpha \times b^\beta \times \dots \times z^\zeta \quad [Hz] \quad (3)$$

Here, a , b , and, z are pitch values, and α , β , and, ζ are blending rates.

For musical key estimation, chord estimation, and drum pattern histogram calculation, the weight in a weight sum corresponds to the blend rates for each song. Note that the sum of the blend rates totals 1.

4.7 Extrapolation song

In the above, we have described a method to generate an in-between song, which corresponds to an interpolation of songs. MusicMean can also generate an extrapolation song, in which a factor of song B can be subtracted from song A. The process for generating an extrapolation song is simple. By setting blend rate to $\alpha > 1.0$ or $\alpha < 0.0$, the system can calculate an external melody and external rhythm pattern.

5. RESULTS AND CONCLUSION

MusicMean allows users to create personalized songs from a selection of songs and blend rate tuning. A song created using the proposed system preserves the characteristics of the original songs. This means that if the user wants to preserve the essence of one song while integrating another song, they can achieve this by manipulating the blend rate by moving a sliding bar. For example, the user can make

rock music quieter by blending ballads. In addition, an average song for a specific musician or album can be generated with MusicMean. The evaluation of the in-between songs and the system is our future work. Early reactions to the proposed system are indicating that users can feel the essence of each original song. However, generated songs often sound strange so that the improvement of the generation quality is a important subject that we have to tackle. In the future, we plan to further analyze in-between songs as well as the evaluation.

Note that an in-between song produced by MusicMean demonstrates some common traits. For example, even after tonality is considered, many minor notes are generated by averaging operation which is not so common in human generated songs. Therefore, MusicMean tends to generate strange melodies. We are planning to solve this problem by considering constraint of more detailed music theory. Another trait is that the melody of an in-between song tends to be flat when mixing many songs because the average converges when many samples are considered. In this case, a statistical model such as HMM based music generation [15] may be better suited for this purpose than the MusicMean fusion approach.

The concept of an average song and an in-between song has great potential for user-personalized music generation. Note that the proposed system represents only an initial phase of our research. We believe that there may be a better method to generate an average song. For example, modulating the keys of an original song, which was not considered in the current study, may be useful for generating a more effective average song. The proposed approach preserves the mood of an original song well; however, there may also be better ways to achieve this. Further the exploration of the averaging method and ongoing development of the proposed system are planned for future work.

As people change the flavor of a dish to their own taste by seasoning, digital content can be equally flexible. MusicMean demonstrates a potential to lead us to next-generation content production and music consumption.

Acknowledgments

We thank Tsukasa Fukusato and Hayato Ohya (Waseda University, Japan) for their advisory about musical theory for MusicMean. This work was supported by IPA Exploratory IT Human Resources Development and partially supported by OngaCREST, CREST, JST

6. REFERENCES

- [1] T. Blaine and S. Fels: Collaborative musical experiences for novices, *Journal of New Music Research*, 32.4, 2003, pp.411–428.
- [2] A. Andrea, and D. Ghisi: A Max Library for Musical Notation and Computer-Aided Composition, *Computer Music Journal*, 2015, pp.11–27.
- [3] J.B. Maxwell, A. Eigenfeldt, and P. Pasquier: ManuScore: Music Notation-Based Computer Assisted Composition, Ann Arbor, MI: MPublishing, University of Michigan Library, 2012.
- [4] J. Bresson, M. Stroppa, and C. Agon: Symbolic Control of Sound Synthesis in Computer-Assisted Composition, in *Proc. of ICMC*, 2005, pp.303–306.
- [5] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue: Computer-assisted composition at IRCAM: from PatchWork to OpenMusic, *Computer Music Journal*, 23.3, 1999, pp.59–72.
- [6] M. Hamanaka, K. Hirata, and S. Tojo: Melody morphing method based on GTTM, in *Proc. of ICMC*, 2008, pp.155–158.
- [7] F. Lerdahl, and R. Jackendoff: An overview of hierarchical structure in music, *Music Perception*, 1983, pp.229–252.
- [8] R.W. Wooller and A.R. Brown: Investigating morphing algorithms for generative music, *Third Iteration: Third International Conference on Generative Systems in the Electronic Arts*, 2005.
- [9] D. Oppenheim: Demonstrating MMorph: a system for morphing music in real-time, in *Proc. of ICMC*, 1995.
- [10] M. D. Hoffman, P. R. Cook, and D. M. Blei: Bayesian spectral matching: Turning Young MC into MC Hammer via MCMC sampling, in *Proc. of ICMC*, 2009.
- [11] T. Hirai, H. Ohya, and S. Morihisa: Automatic Mash up Music Video Generation System by Perceptual Synchronization of Music and Video Features, in *Proc. of SIGGRAPH* posters, 2012.
- [12] T. Naknao, S. Murofushi, M. Goto, and S. Morihisa: DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Video Clips on the Web, in *Proc. of SMC*, 2011, pp.183–189.
- [13] M. E. P. Davies, P. Hamel, K. Yoshii and M. Goto: AutoMashUpper: Automatic creation of multi-song mashups, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(12), 2014, pp. 1726–1737.
- [14] J. Sneyers, and S. D. Danny: APOPCALEAPS: Automatic music generation with CHRiSM, in *Proc. of ISMIR*, 2010.
- [15] D. Conklin: Music generation from statistical models, in *Proc. of AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, pp.30–35.