TOWARDS EXPRESSIVE INSTRUMENT SYNTHESIS THROUGH SMOOTH FRAME-BY-FRAME RECONSTRUCTION: FROM STRING TO WOODWIND

Sanna Wager, Liang Chen, Minje Kim, and Christopher Raphael

Indiana University
School of Informatics and Computing
Bloomington, IN, USA 47408
{scwager, chen348, minje, craphael}@indiana.edu

ABSTRACT

We consider the task of mapping the performance of a musical excerpt on one instrument to another. Our focus is on excitation-continuous instruments, where pitch, amplitude, spectrum, and time envelope are controlled continuously by the player. The synthesized instrument should follow the target instrument's expressive gestures as much as possible, while also following its natural characteristics. We develop an objective function that balances distance of the synthesis to the target and smoothness in the spectral domain. An experiment mapping violin to bassoon playing by concatenating together short excerpts of audio from a database of solo bassoon recordings serves as an illustration.

Index Terms— Concatenative Sound Synthesis, Instrumental Synthesis

1. INTRODUCTION

Applications such as information retrieval using audio queries [1], source separation by humming/singing [2], and humming/singing-to-instrument synthesis benefit from the ability to synthesize melodies, which can be done using Concatenative Sound Synthesis (CSS, concatenation of short audio samples to match a target performance or score). Such synthesis—extensively explored for voice [3][4]—faces challenges when the desired expressive parameters change quickly or subtly, or have a wide range as occurs in musical genres such as classical or jazz. Even mild discontinuity in the spectral domain is often audible and displeasing to the listener: Synthesized performances of melodies requiring refined control of expressive parameters are likely to have such glitches. We address this challenge using a variant of CSS to synthesize melodies on instruments such as strings, voice, or winds where expressive parameters—pitch, amplitude, spectrum, and time envelope—are continuously controlled.

Sample-based CSS addresses spectral discontinuity by concatenating long-duration samples—full or half-notes— and substantially post-processing the results in the spectrum, amplitude, pitch, and expression [5][6][7][8][3]. Concatenation of longer excerpts risks discontinuity at the broader level of the expressive gesture; and the post processing that can be applied without unnatural results is limited, especially in the case of instruments such as the bassoon, where spectrum, onset and time envelope shapes vary abruptly at different pitches and dynamic levels. A related method, audio mosaicing [9][10], deploys samples that are windows of a fixed number of milliseconds, increasing expressive flexibility at the gesture level, but with frequent spectral discontinuity and less retention of the expressive characteristics of the source instrument.

We combine the realism of sample-based CSS with the expressive flexibility of audio mosaicing. Like audio mosaicing, we treat every window of 12ms as a sample. However, we favor selection of consecutive windows to increase continuity, and even force it during note changes, which are particularly delicate transitions. Additionally, we only allow concatenations of nonconsecutive frames that are measured as "similar" in pitch, timbre and amplitude. The need for post processing is substantially reduced: The probability of finding an appropriate short sequence of frames to match a sequence of target frames is higher than that of being able to match a full note.

Our work builds on two algorithms: 1) the audio mosaicing technique inspired by Nonnegative Matrix Factorization [11][12] that selects consecutive database frames [9], and 2) "Infinite Jukebox" algorithms [13] that make a popular tune last arbitrarily long by building a graph that identifies appropriate transitions between similar-sounding beats using randomly chosen transition paths. We adapt the concept of a graph to continuously-controlled instruments, where parameters such as pitch, amplitude, and spectral shape change fluidly at the timescale of milliseconds instead of beats.

We demonstrate our approach by mapping a performance on a string instrument (violin) to a wind instrument (bassoon). The challenge is to retain the musical gesture of the target without changing the characteristics of the source. Vibrato on a string instrument, for example, depends mainly on changes in pitch and tends to be rapid, whereas vibrato on a wind instrument is slower and depends more on timbre and loudness than on pitch. We develop initial familiarity with the method using a nearest-neighbor search between source and target, then explore nonparametric approaches such as regression trees [14].

2. THE PROPOSED MODEL

Our model uses two criteria to optimize a sequence of source frames. We first ensure that transitions between non-consecutive frames are smooth in the spectral domain by building a graph that connects every source frame to all those to which it is similar in spectrum. Transitions are only allowed between connected frames. We then make the sequence of source frames match the expressive gestures of the target instrument by minimizing expressive distance between source and target at each frame. Finally, we assemble the selected sequence into a new recording using the phase vocoder.

2.1. Features

We segment the audio into frames of 12ms, storing the following features for each frame: nominal pitch (MIDI pitch ranging from 0

to 127), fundamental frequency, the modulus of the windowed Short-Time Fourier Transform (STFT), and energy measured as the Root-Mean-Square (RMS):

$$RMS = \sqrt{\sum_{n=0}^{N-1} Win[n] * x[n]^2}$$

where Win is a Hann window of length N. RMS values depend on recording settings and may differ from pitch to pitch on a given instrument, thus each MIDI pitch in the target and the database is treated as having its own normal RMS distribution. The means and standard deviations are smoothed using linear regression across the full range of pitches, with individual RMS values stored as quantiles.

2.2. Cost 1: Target-to-source mapping

Expressive similarity between target and source is a hard-to-define concept. An ideal distance measure would capture the features that are common to all instruments, such as pitch and amplitude trajectories, and ignore instrument-specific ones like spectrum, vibrato rate, and time envelope. In practice, the two are hard to distinguish. For example, vibrato directly affects pitch and amplitude. We chose to base the distance metric between two STFT frames i and j, $d_1(i,j)$[1]. The metric is defined as a weighted sum of the difference in their fundamental frequencies $\bar{F}_i$ and $\bar{F}_j$ (in Hz) and that of RMS-quantiles Q_i and Q_j for a given pair of frames. We transpose the target pitch measurements to match the range of the source instrument.

$$d_1(i,j) = |\bar{F}_i - \bar{F}_j| + w \times |Q_i - Q_j| + C \tag{1}$$

where w is a weighting constant. For the later use, we define another variable $\bar{f}_i$, which is the frequency bin index of $\bar{F}_i$. C is an additional cost used to avoid any accidental discrepancies between the two nominal pitches N_i and N_j defined as follows:

$$C = \begin{cases} 0 & \text{if } N_i = N_j \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

The fundamental frequencies $\bar{F}_i$ are found by the YIN algorithm [15], but sometimes the result can be noisy. To fix this, we rely on their nominal pitch from the aligned score. If the ratio between estimated fundamental frequency $\bar{F}_i$ and the nominal pitch N_i is above a threshold, e.g. $\bar{F}_i/N_i > 1.025$ or $\bar{F}_i/N_i < 0.975$ we replace $\bar{F}_i$ with $\bar{F}_{i-1}$, with one exception: if the i-th frame is an onset frame we replace $\bar{F}_i$ with N_i. Our next step will be to incorporate the pYin [16] algorithm to smooth results.

2.3. Cost 2: Database transition graph

In addition to target-to-source mapping, which lacks the concern about the local smoothness among the recovered frames, we employ another cost that controls the continuity of the participating source frames. To this end, we construct a transition graph from the pairwise similarity between the database frames.

An ideal transition graph connects the database frames that are either consecutive or similar enough to each other that audio can be connected at these points without causing noticeable discontinuity in spectrum, pitch, or amplitude at the frame-to-frame level. Three choices are possible when selecting a sequence of database frames for the synthesis: 1) extend the current sample of bassoon by continuing with the next frame in the database; 2) repeat the current frame to increase the duration of the current-frame sound; and 3) as in the Infinite Jukebox [13], "jump" to any frame connected to the current one in the database graph, thus ending the current sample and concatenating a new one to it. A sequence of consecutive frames can last anywhere from one to hundreds of frames, breaking when this is necessary for the reconstruction to match the target.

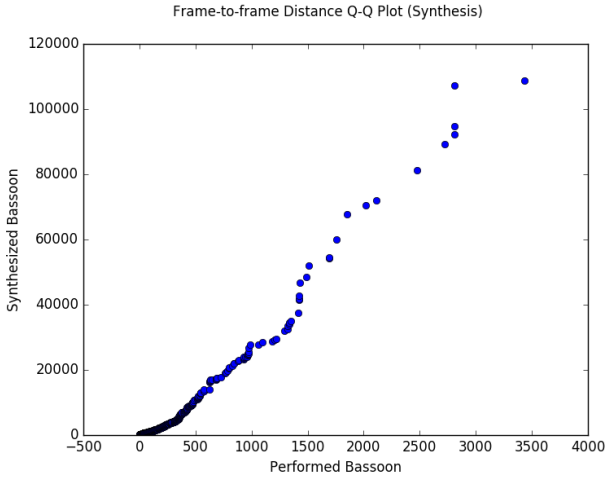


Fig. 1: Q-Q plot of the frame-to-frame distances $d_2(i,j)$ of a bassoon performance of the Sibelius excerpt and of the proposed bassoon synthesis.

We measure frame-to-frame distance as the Euclidean distance of the windowed STFT modulus of the neighborhoods of the k first partials, with frames whose distance from each other is less than a selected threshold designated as connected in the graph. The distance d_2 between the i-th and j-th database frames is defined as follows:

$$d_2(i,j) = \|\boldsymbol{h}_{(i,j)} - \boldsymbol{h}_{(j,i)}\|_2, \tag{3}$$

where we define $\boldsymbol{h}_{(i,j)} \in \mathbb{R}^{2K}$ as a set of summed neighboring Fourier magnitudes around K harmonic peaks from i-th and j-th frames, respectively. For the first K harmonic peaks of i-th frame, we first sum the magnitudes of its c neighboring bins,

$$\boldsymbol{h}_{(i,j)}(k) = \sum_{f=k\bar{f}_i-c}^{k\bar{f}_i+c} |\boldsymbol{x}_i(f)|, \quad k = \{1, \cdots, K\} \tag{4}$$

where $\boldsymbol{x}_i(f)$ denotes the f-th frequency bin of a Fourier spectrum for the i-th frame, and $k\bar{f}_i$ is the bin index of the k-th harmonic partial. Furthermore, for the second half of its elements we also gather values from the bins associated with the harmonics of the j-frame:

$$\boldsymbol{h}_{(i,j)}(k+K) = \sum_{f=k\bar{f}_j-c}^{k\bar{f}_j+c} |\boldsymbol{x}_i(f)|, \quad k = \{1, \cdots, K\} \tag{5}$$

$\boldsymbol{h}_{(j,i)}$ is defined in a similar way, except we collect values from $\boldsymbol{x}_j$.[2]

[1] In this section we assume that i and j are from the source and the target instruments, respectively

[2] This distance measure gave better results than cosine distance of the STFT partials, Euclidean or cosine distance of the Constant-Q Transform (CQT), or the RMS difference.

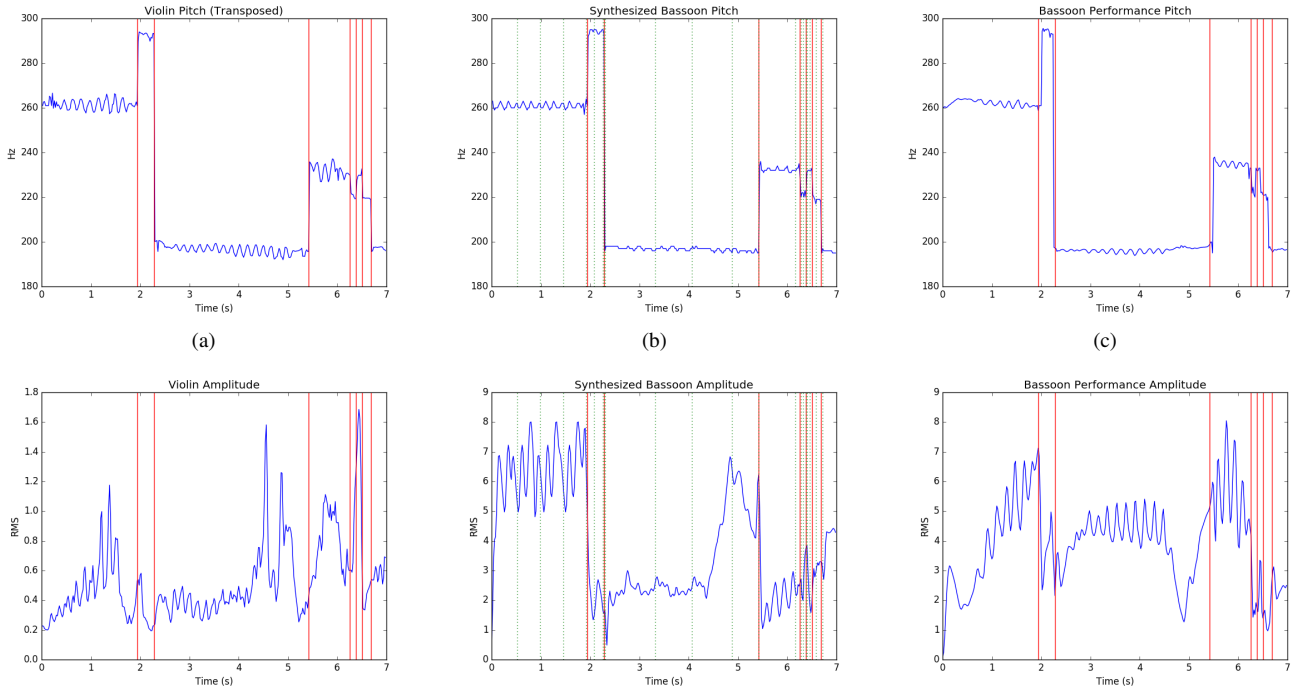


Fig. 2: Pitch and amplitude trajectories of the target (violin), the proposed bassoon synthesis, and the musical excerpt performed on the bassoon over the 7 first seconds of audio. Solid lines indicate note changes in the score, while dotted lines indicate where non-consecutive frames were selected for the synthesis. The sound waves were aligned in time using a dynamic time warping algorithm and normalized to have the same average loudness.

The graph makes transitions between separate audio samples smooth at the frame-to-frame level but does not prevent discontinuity at the level of a note or a musical phrase. A transition can occur in the middle of a vibrato cycle, for example, truncating it and causing it to lose its contour. Smoothness at this higher level depends on the target-to-source mapping quality.

2.4. Objective Function

A global cost $\mathcal{J}$, constructed by summing the source-to-target frame distances and source-to-source transitional penalties, is minimized subject to the constraint of transitions allowed by the graph:

$$\arg\min_{\mathbf{v}\in\mathcal{V}} \mathcal{J} = \arg\min_{\mathbf{v}\in\mathcal{V}} \sum_{i=1}^{T} d_1(i, \mathbf{v}_i) + P(\mathbf{v}_{i-1}, \mathbf{v}_i), \qquad (6)$$

where the T-dimensional index vector, $\mathbf{v}$, is a sequence of candidate database frames, whose i-th element points to one of S database frames for its corresponding i-th target frame. Since there are S total frames in the source database, the set of paths $\mathcal{V}$ contains exponentially (T^S) many candidate sequences we can choose from during the minimization procedure. The second term P gives penalty to less favorable transitions to reduce the search space:

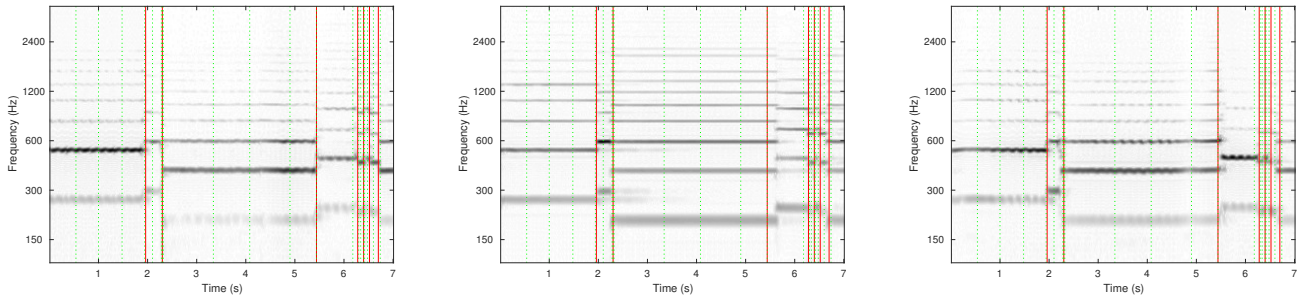
$$P(\mathbf{v}_{i-1}, \mathbf{v}_i) = \begin{cases} 0 & \text{if } \mathbf{v}_{i-1} + 1 = \mathbf{v}_i \\ \alpha_1 & \text{if } \mathbf{v}_{i-1} = \mathbf{v}_i \\ \alpha_2 & \text{if } d_2(\mathbf{v}_{i-1}, \mathbf{v}_i) < \tau \\ & \text{and } \mathbf{v}_{i-1} + 1 \neq \mathbf{v}_i \\ & \text{and } \mathbf{v}_{i-1} \neq \mathbf{v}_i \\ \infty & \text{otherwise} \end{cases} \qquad (7)$$

Transitions where the database frame for the i-th target frame $\boldsymbol{v}_i$ is dissimilar enough to that of the preceding target frame $\boldsymbol{v}_{i-1}$ that the difference exceeds τ are assigned a transition distance of infinity. When the selected source frames for a consecutive pair of target frames happen to be consecutive in the source signal as well, we do not add any penalty. We add α_1 when repeating a source frame, as excessive repetition of source frames sounds unnatural. We add α_2 when two adjacent recovered frames are neither adjacent nor same in the database because distance risks harming the smoothness. The objective function (6) is optimized using the Viterbi algorithm [18] subject to a hard constraint: database frames which occur in a "note change region"—defined as the range of frames before or after a note change—can be selected only when the target is also in a "note change region".

3. EXPERIMENTS

We selected as target the opening of the Sibelius violin concerto, performed by a musician from the Indiana University Jacobs School of Music, for its long legato notes that are connected and played smoothly. The database consists of approximately 13 minutes of bassoon playing by a professional bassoonist[3], recorded in the same room for consistency of audio quality. Pieces performed by the bassoonist were chosen to have long, connected notes like the target piece, while not including the target melody. All recordings were parsed to match a MIDI score using the Music Plus One program [19]. The audio settings were as follows: sampling rate 48000 Hz, frame length 4096, and hop length 1024. The STFT was computed using the LibROSA package [20], applying an asymmetric Hann

[3]The authors would like to express their gratitude to Professor Kathleen McLean from the Indiana University Jacobs School of Music



(a) Proposed (b) MIDI (c) Garage Band Synthesis

Fig. 3: Constant Q transform of the synthesized bassoon (proposed), a MIDI synthesis generated using GarageBand [17], and the bassoon performance over the 7 first seconds of audio.

window to each frame. The fundamental frequency was computed using the YIN algorithm and manually corrected for errors.

3.1. Parameter Settings

Parameters for the model were derived empirically. Weighting constant w for the target-to-source distance $d_1(i, j)$, described in section 2.2, was set to 50. In the database transition graph distance metric of section 2.3, $K = 7$ and $c = 2$ gave the best results when measuring the quality as the count of "smooth" connections (as judged by the authors) among a sample of 50 randomly generated frame connections under the given settings. In the objective function of section 2.4, setting $\tau = 39$ for the transitional penalty gave best results. Penalty values were set to $\alpha_1 = 70$ and $\alpha_2 = 100$, which favored a limited number of frame repetitions. The size of the "note change region", where no non-consecutive transitions are allowed, was set to 10 frames after examining the behavior of the source data at note boundaries.

3.2. Evaluation and Results

We examine how much the synthesis follows the target's expressive gestures and how well it preserves source instrument characteristics. Thus, we compare the bassoon synthesis both to the violin performance and to a recording of the same bassoonist performing the Sibelius excerpt, imitating the violin's expressive gestures[4]. We aligned the three recordings in time using dynamic time warping, and normalized them to have the same average loudness. Figure 1 compares frame-to-frame distances $d_2(i, j)$ of the bassoon performance and proposed bassoon synthesis using a Q-Q plot and shows substantial similarity in the distributions. Figure 2 displays the pitch and amplitude (RMS) trajectories of the beginnings of the three recordings. Figure 3 shows Constant Q transform spectral features of the synthesis, bassoon performance, and a Garage Band-synthesized performance [17] for comparison with a standard synthesis method. Visual inspection shows that the synthesized bassoon has a mixture characteristics of the violin and bassoon performances. Generally, the synthesized bassoon seems to follow the contour of the violin while retaining some of its natural characteristics. However, some instrument-specific characteristics cause small glitches. Vibrato is an example. The violin performance vibrato on the first note starts at the onset rather than developing gradually as in the bassoon performance, is wider throughout, and has a faster rate. We observe that synthesized bassoon has an even vibrato throughout instead of growing over time, probably because the widest vibrato a bassoon can produce was consistently selected to match the violin. The fact that a wide vibrato is usually played at a louder dynamic explains the high amplitude of the synthesis at the beginning of the excerpt compared to both performances. The occasional angularity that can be seen in both the amplitude and pitch, and can correlate with "wobbles" in the sound, may have emerged when the target-to-source mapping caused the bassoon to attempt to imitate the faster rate of violin vibrato, potentially causing non-consecutive frame concatenations in the middle of bassoon vibrato cycles. Instrument-specific melodic behavior serves as a second example. The latter part of the recording has large leaps in the violin, which are rare on the bassoon. As expected, the lesser representation of such transitions in the bassoon database makes the synthesis of these passages sound less smooth.

[4]Recordings of the target, the synthesis, and, for comparison, of the bassoonist performing the target while imitating the musical gestures of the target can be found at `http://homes.soic.indiana.edu/scwager/css.html`

4. CONCLUSIONS AND FUTURE WORK

Our model concatenates a sequence of source frames with optimized smooth frame-to-frame transitions and minimizes the distance between full sequence, and target, expressive gestures. We will explore elimination of the discontinuities at the level of the expressive gesture that remain using a nonparametric or data-driven refinement of the target-to-source distance metric, or additive synthesis using neural networks [21]. Such developments to the model would reduce the number of parameters which need to be hand-tuned. A key motivation for our model was to reduce the amount of post processing required for the pitch, amplitude and spectrum and time envelope to change smoothly over time, in order keep the sound as natural as possible. Our results—that exclude post processing—encourage us to explore limited and subtle post-processing to further smooth the results.

This model was designed for a very specific context, and is thus limited in its scope. It requires consistent recording settings: use of microphones with different frequency responses may cause a continuously high $d_1(i, j)$, and differing levels of interfering noise and reverberation will decrease model reliability. Making the model robust to changes in recording setting—for example, via pre-processing—would make it possible to use data from different performers. Furthermore, the model depends on knowledge of the score, but can be further developed for the situation where no score is available, in order to be generalizable to contexts such as humming-to-instrument synthesis.

5. REFERENCES

[1] Y. Zhang and Z. Duan, "Imisound: An unsupervised system for sound query by vocal imitation," in *ICASSP*, 2016, pp. 2269–2273.

[2] P. Smaragdis and G. J. Mysore, "Separation by humming: user-guided sound extraction from monophonic mixtures," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 69–72.

[3] M. Goto, T. Nakano, S. Kajita, Y. Matsusaka, S. Nakaoka, and K. Yokoi, "VocaListener and VocaWatcher: Imitating a human singer by using signal processing," in *ICASSP*, 2012, pp. 5393–5396.

[4] J. Bonada, A. Loscos, and H. Kenmochi, "Sample-based singing voice synthesizer by spectral concatenation," in *Proceedings of Stockholm Music Acoustics Conference*, 2003, pp. 1–4.

[5] E. Maestre, R. Ramírez, S. Kersten, and X. Serra, "Expressive concatenative synthesis by reusing samples from real performance recordings," *Computer Music Journal*, vol. 33, no. 4, pp. 23–42, 2009.

[6] B. L. Sturm, "Adaptive concatenative sound synthesis and its application to micromontage composition," *Computer Music Journal*, vol. 30, no. 4, pp. 46–66, 2006.

[7] D. Schwarz and B. Hackbarth, "Navigating variation: composing for audio mosaicing," in *International Computer Music Conference (ICMC)*, 2012, pp. 1–1.

[8] D. Schwarz, "The caterpillar system for data-driven concatenative sound synthesis," in *Digital Audio Effects (DAFx)*, 2003, pp. 135–140.

[9] J. Driedger, T. Prätzlich, and M. Müller, "Let it bee - towards NMF-inspired audio mosaicing," in *ISMIR*, 2015.

[10] A. Lazier and P. Cook, "Mosievius: Feature driven interactive audio mosaicing," in *Digital Audio Effects (DAFx)*, 2003.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[13] P. Lamere, "The infinite jukebox," https://musicmachinery.com/2012/11/12/the-infinite-jukebox/.

[14] D. Stowell and M. D. Plumbley, "Timbre remapping through a regression-tree technique," *Sound and Music Computing (SMC)*, 2010.

[15] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[16] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *ICASSP*, 2014, pp. 659–663.

[17] Apple Inc., "Garage Band (music editing software)," http://www.apple.com/mac/garageband/, 2016.

[18] L. R. Rabiner, "Readings in speech recognition," chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296. 1990.

[19] C. Raphael, "Music plus one and machine learning," in *ICML*, 2010, pp. 21–28.

[20] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015.

[21] Eric Lindemann, "Music synthesis with reconstructive phrase modeling," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 80–91, 2007.