Zafar Rafii

The Constant-Q Harmonic Coefficients

A timbre feature designed for music signals

Timbre is the attribute of sound that makes, for example, two musical instruments playing the same note sound different. It is typically associated with the spectral (but also the temporal) envelope and assumed to be independent from the pitch (but also the loudness) of the sound [1]. This article shows how to design a simple but effective pitch-independent timbre feature, well adapted to musical data, by deriving it from the constant-Q transform (CQT), a log-frequency transform that matches the typical Western musical scale [2], [3]. The decomposition of the CQT spectrum into an energy-normalized pitch component and a pitch-normalized spectral component is demonstrated, the latter from which a number of harmonic coefficients are extracted. The discriminative powers of these constant-Q harmonic coefficients (*CQHCs)* are then evaluated on the NSynth data set [4], a publicly available, large-scale data set of musical notes, where they are compared with the mel-frequency cepstral coefficients (MFCCs) [5], a feature originally designed for speech recognition but commonly used to characterize timbre in music.

Relevance

A timbre feature that is well adapted to musical data, pitch independent, and that has good discriminative power can find uses in a number of applications, such as similarity detection, sound recognition, and audio classification, in particular, of musical instruments. Additionally, the ability to decompose the spectrum of a sound—here, the CQT spectrum—into a pitch-normalized spectral component and an energy-normalized pitch component can be useful for the analysis, transformation, and resynthesis of music signals. An energy-normalized pitch component can also potentially be used for tasks such as pitch identification and melody extraction.

Prerequisites

A basic knowledge of audio signal processing and some knowledge of music information retrieval (MIR) [6] are required to understand this article, in particular, concepts such as the Fourier transform (FT), CQT, MFCCs, convolution, and pitch.

Problem statement

The multidimensional nature of timbre makes it an attribute that is tricky to quantify in terms of one simple characteristic feature [7]. Although it is assumed to be independent from pitch and loudness, it is not really feasible to fully disentangle timbre from those qualities as timbre is inherently dependent on the spectral content of the sound, which is also defined by its pitch and loudness [1]. Researchers in MIR proposed a number of descriptors to characterize one or more aspects of timbre [8], but they frequently resort to using the MFCCs when they need one simple timbre feature [6]. Although the MFCCs were shown to be practical in a number of MIR tasks, they were initially designed for speech processing applications [5] and are not necessarily well adapted to musical data. In particular, they are derived through an old procedure that makes use of the mel scale, a perceptual scale experimentally designed 85 years ago to approximate the human auditory system's response [9]. More recently, a number of data-driven approaches attempted to learn some timbral representations from musical data, but typically in terms of implicit embeddings, which are tied to specific trained models [4], [10]–[13] and not necessarily as explicit and interpretable features such as the MFCCs, which are still often preferred as the simple go-to feature to characterize timbre in music by MIR practitioners.

Although the MFCCs were shown to be practical in a number of MIR tasks, they were initially designed for speech processing applications and are not necessarily well adapted to musical data.

Solution

The CQHCs is a timbre feature that is well adapted to musical data,

Digital Object Identifier 10.1109/MSP.2021.3138870
Date of current version: 27 April 2022

 1053-5888/22©2022IEEE

pitch-independent, simple to compute, interpretable, and effective. First, a brief description of the CQT is presented, then it is shown how one can decompose the CQT spectrum into a pitch-normalized spectral component and an energy-normalized pitch component, and finally extract a number of coefficients from the spectral component. Note that this work does not claim to propose the best timbre feature for MIR applications, as today's data-driven approaches are able to learn decent timbral representations, but rather to illustrate how to design a good timbre feature without relying on any form of supervised learning, using common theories from digital signal processing.

Description of the CQT

The CQT is a frequency transform with a logarithmic resolution that matches the equal temperament, a tuning system typically used in Western music that divides an octave into equal steps [2]. Unlike the discrete Fourier transform (DFT), which uses the same window length for every frequency bin, leading to a linear frequency resolution, the CQT uses window lengths that decrease with increasing frequency, with a ratio of center frequency to frequency width—also known as the *quality factor* (Q)—which is constant, leading to a logarithmic frequency resolution where the frequency bins essentially correspond to the tones in the typical Western musical scale, given a chosen octave resolution (i.e., the number of frequency bins per octave). Although the direct calculation of the CQT is slow, a fast implementation was also proposed, which makes use of the fast Fourier transform (FFT) in conjunction with the use of a kernel [3].

The logarithmic frequency resolution of the CQT allows the harmonics of musical notes to form a constant pattern in the frequency domain, with the relative positions of the harmonics remaining the same as the fundamental frequency shifts up or down in frequency. The locations of these harmonics in the CQT spectrum (i.e., the magnitude or power of the original complex CQT) therefore depends only on the location of the fundamental frequency and the chosen octave resolution. As harmonics are the spectral coefficients that carry most of the instrument's information, they are a good candidate to characterize the timbre of the instrument. Provided that the CQT spectrum can be normalized in pitch, i.e., by somehow bringing the fundamental frequency down to the lowest frequency bin, the locations of the harmonics could then be easily inferred and their energy could be extracted, leading to a simple but effective timbre feature for music signals.

> **The CQHCs is a timbre feature that is well adapted to musical data, pitch-independent, simple to compute, interpretable, and effective.**

Deconvolution of the CQT spectrum

To show how to "pitch normalize" a CQT spectrum, first, the assumption is that a CQT spectrum X can be represented as the convolution between a pitch-normalized spectral component S (which mostly contains the timbre information) and an energy-normalized pitch component P (which mostly contains the pitch information), as shown in (1), where $*$ represents the convolution operation.

$$X = S * P \tag{1}$$

This convolution process can also be thought of as a source-filter model [14], which is here not applied in the time domain but rather the frequency domain, with the source and the filter being the pitch and spectral components, respectively.

Observation 1: A pitch change in the audio translates to a linear shift in the CQT spectrum [2], [3]. Assuming that pitch and timbre are independent, this implies that the same musical object at different pitches would have a similar spectral component but a shifted pitch component (while two different musical objects at the same pitch would have different spectral components but a similar pitch component). This is summarized in (2), where X, S, P, and X', S', P' represent the CQT spectrum, spectral component, and pitch component for a musical object, and for a pitch-shifted version of the same musical object, respectively, and $\approx$ represents the approximate equality.

$$\begin{cases} X = S * P \\ X' = S' * P' \end{cases} \\ \Rightarrow S \approx S' \tag{2}$$

Note that, although in theory the equality should be exact, in practice, two musical objects at two different pitches (e.g., two different notes from the same instrument) are not exact translations of each other, hence the approximate equality.

Observation 2: The FT of the convolution between two functions is equal to the pointwise product between the FTs of the two functions, a property known as the convolution theorem [15]. This implies that the FT of the CQT spectrum is equal to the pointwise product between the FT of the spectral component and the FT of the pitch component. Given the first observation, this further implies that the FT of the spectral component for a musical object and for a pitch-shifted version of it would be equal. This is summarized in (3), where $\mathcal{F}(.)$ represents the FT function and $\cdot$ the pointwise product.

$$\begin{cases} \mathcal{F}(X) = \mathcal{F}(S * P) = \mathcal{F}(S) \cdot \mathcal{F}(P) \\ \mathcal{F}(X') = \mathcal{F}(S' * P') = \mathcal{F}(S') \cdot \mathcal{F}(P') \end{cases} \\ \Rightarrow \mathcal{F}(S) \approx \mathcal{F}(S') \tag{3}$$

Observation 3: The magnitude FT is shift invariant [15]. This implies that the magnitude of the FT of the CQT spectrum for a musical object and for a pitch-shifted version of it would be equal. This is summarized in (4), where $|.|$ and $Arg(.)$ represent the modulus and argument, respectively, for a complex array and j, the imaginary unit.

$$\begin{cases} \mathcal{F}(X) = \left|\mathcal{F}(X)\right| \cdot e^{jArg(\mathcal{F}(X))} \\ \mathcal{F}(X') = \left|\mathcal{F}(X')\right| \cdot e^{jArg(\mathcal{F}(X'))} \end{cases} \\ \Rightarrow \left|F(X)\right| \approx \left|F(X')\right| \tag{4}$$

Given the previous observations, we can therefore infer that the FT of the spectral component could be approximated by the magnitude FT of the CQT spectrum, while the FT of the pitch component could be approximated by the phase component. This finally gives us the estimates for the spectral component and the pitch component, after taking their inverse FTs, as shown in (5), where $\mathcal{F}^{-1}(.)$ represents the inverse FT function.

$$\Rightarrow \begin{cases} \mathcal{F}(S) \approx |\mathcal{F}(X)| \\ \mathcal{F}(P) \approx e^{jArg(\mathcal{F}(X))} \end{cases}$$
$$\Rightarrow \begin{cases} S \approx \mathcal{F}^{-1}(|\mathcal{F}(X)|) \\ P \approx \mathcal{F}^{-1}(e^{jArg(\mathcal{F}(X))}) \end{cases} \quad (5)$$

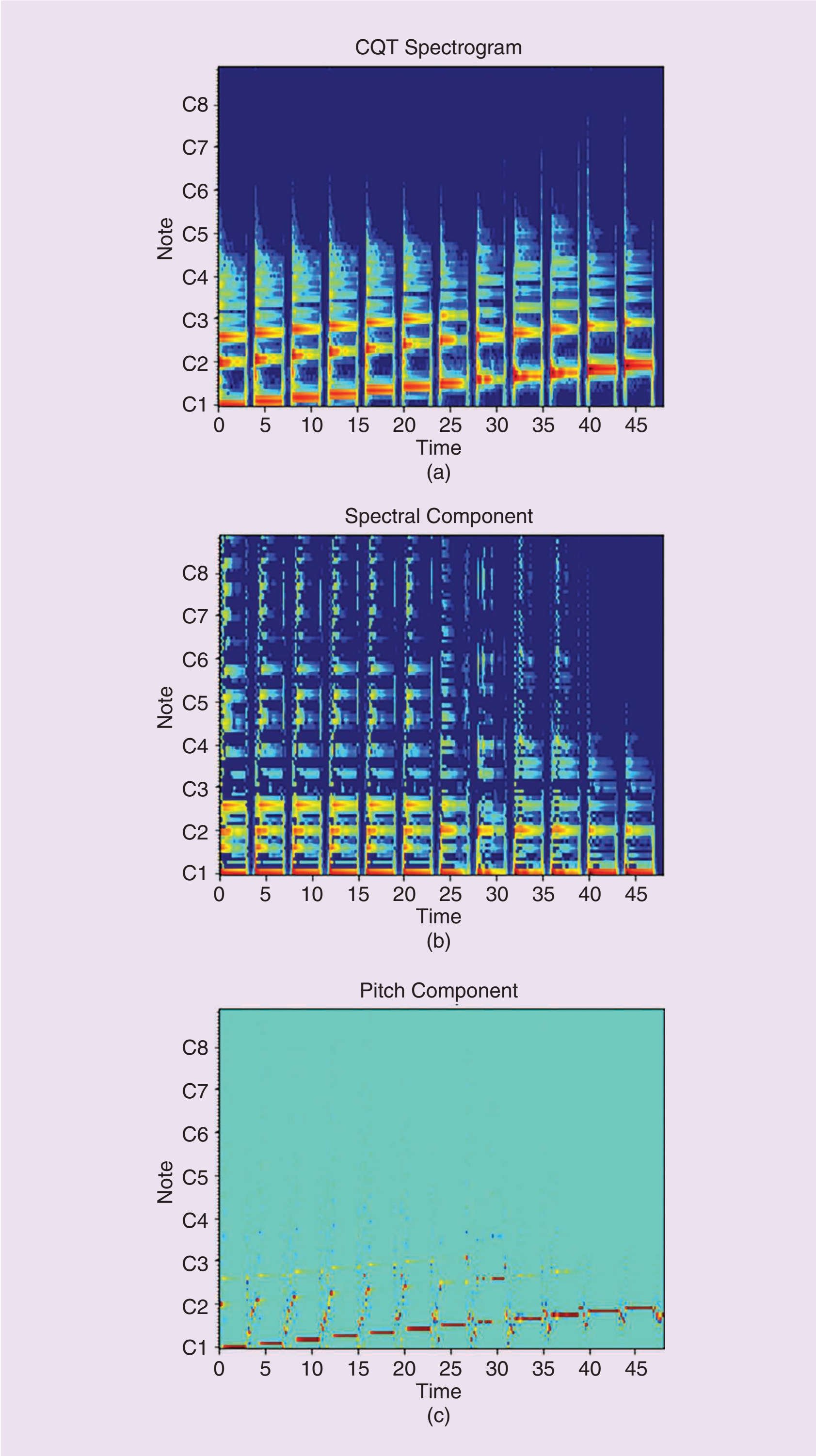

FIGURE 1. (a) The deconvolution of the CQT spectrogram (shown in dB) of 12 acoustic bass notes playing from C1 to B1, into (b) a pitch-normalized spectral component (shown in dB), and (c) an energy-normalized pitch component (shown in [0, 1]).

Figure 1 depicts an example of the deconvolution of a CQT spectrogram (i.e., a concatenation of CQT spectra over time) into a pitch-normalized spectral component and an energy-normalized pitch component. The CQT spectrogram was computed from an audio signal created by concatenating 12 4-s notes of an acoustic bass playing from C1 (32.70 Hz) to B1 (61.74 Hz) in ascending pitch. The notes come from the NSynth data set [4] and correspond to instrument id `bass_acoustic_000`, MIDI numbers `024` to `035`, and velocity number `075`. The CQT spectrogram was computed using librosa [16], [17], with a sampling rate of 16 kHz, a hop length of 512 samples, a minimum frequency of 32.7 Hz (corresponding to C1), 95 frequency bins, and 12 bins per octave. As shown in Figure 1, the spectral component looks as if the CQT spectrogram has been normalized in pitch, with all the notes being brought down to the lowest fundamental frequency (corresponding to C1), while the pitch component looks as if the CQT spectrogram has been stripped down from its energy, leaving mostly the fundamental frequencies of the notes. Note that in practice, a power CQT spectrogram (i.e., magnitude to the power of 2) is used and the real part of both the spectral and pitch components are taken to ensure real values. This deconvolution can also potentially be postprocessed, for example, by zeroing the few negative values in the pitch component and using them to derive a refined spectral component.

This deconvolution process can also be thought of as the normalization of the CQT spectrum by the magnitude of its FT (which here would correspond to the FT of the spectral component), leading to a sharper CQT spectrum (which here would correspond to the pitch component) in the manner of the generalized cross-correlation phase transform (GCC-PHAT) method, which aims at normalizing a cross-correlation function by its magnitude spectrum to sharpen the cross-correlation peaks [18].

Note that the authors in [19] presented a method they called *specmurt analysis*, where they similarly assumed that a CQT-like log spectrum can be decomposed into a so-called common harmonic structure and a fundamental frequency distribution, however, with the aim of extracting multiple fundamental frequencies from polyphonic music signals, focusing essentially on the pitch component rather than the spectral one. Inspired by the derivation of the cepstrum [20], they proposed estimating the fundamental frequency distribution through inverse filtering, starting from an initial assumed common harmonic structure and using an iterative algorithm to refine it in order to refine the fundamental frequency distribution itself. Although their final leftover harmonic structure could potentially be used to also extract harmonic coefficients, note that the pitch-normalized spectral component shown in (5) is derived through a simpler and more intuitive idea, which is not based on heuristics or an iterative algorithm.

Extraction of the harmonic coefficients

The spectral component resulting from the deconvolution of the CQT spectrum can thus be thought of as a pitch-normalized CQT spectrum where the harmonics of the instrument have essentially been brought down to the same lowest note level. Given the octave resolution, which was used when computing the CQT, we can then easily infer the locations of those harmonics in the spectral component. We can subsequently extract these harmonic coefficients from the spectral component and thus obtain a compact and interpretable feature for characterizing the timbre of the instrument. Equation (6) demonstrates how to derive the indices of the harmonic coefficients given O_r, the octave resolution, and N_c, the number of desired coefficients, and finally extract the CQHCs from the spectral component S, with $\log_2(.)$ and *round*(.) representing the binary logarithm and the round function, respectively.

A CQT spectrum *X* can be represented as the convolution between a pitch-normalized spectral component *S* (which mostly contains the timbre information) and an energy-normalized pitch component *P* (which mostly contains the pitch information).

$$\begin{cases} i = round(O_r \log_2(k)) \\ \mathrm{CQHC}_k = S(i) \end{cases} \qquad 1 \le k \le N_c \qquad (6)$$

Figure 2(a) displays an example of CQHCs. Twenty coefficients were extracted from the spectral component resulting from the deconvolution of the CQT spectrogram of the musical signal shown in Figure 1. These coefficients essentially correspond to the harmonics of the instrument, which contains most of its spectral energy and can therefore be a reasonable representation of the timbre of the instrument. For comparison, Figure 2(b) shows the MFCCs derived from the same musical signal. Twenty coefficients were computed using librosa [16], with a sampling rate of 16 kHz, a window length of 1,024 samples, and hop length of 512 samples, matching the time resolution of the CQHCs so that both features have the same size in time and frequency.

The reader is reminded how the MFCCs are commonly derived. First, a classic spectrogram is computed from the audio signal using the short-time Fourier transform (STFT), and the powers of the frequencies in each frame are then mapped onto the mel scale [9] using triangular overlapping windows. This essentially produces a mel spectrogram. Then the log of every mel frequency is taken, followed by the discrete cosine transform of each frame. Finally, the MFCCs are selected as the lowest coefficients in the resulting spectrum, excluding the very first one

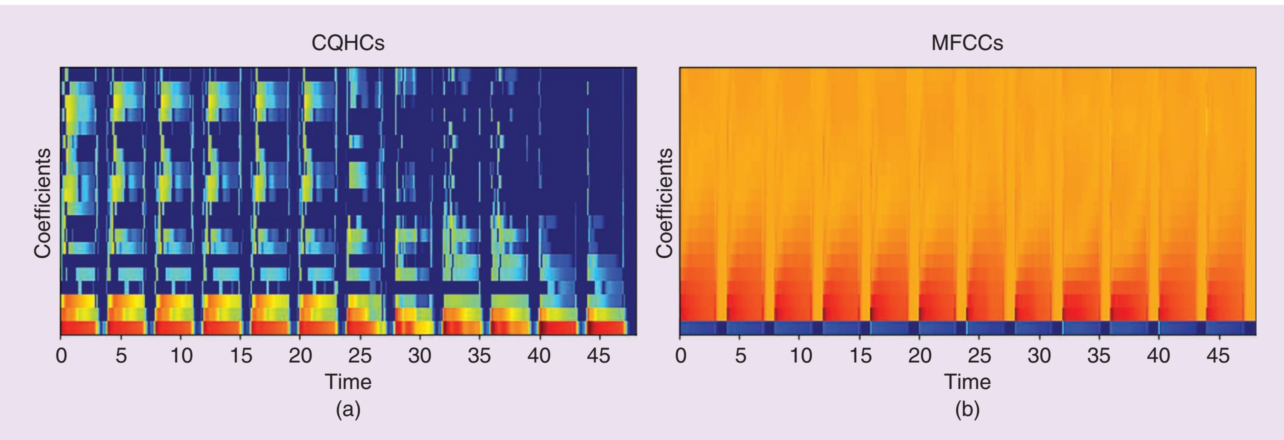

FIGURE 2. (a) The CQHCs extracted from the spectral component obtained following the deconvolution of the CQT spectrogram shown in Figure 1(a) (shown in dB) and (b) the MFCCs computed from the same musical signal.

(i.e., the dc component). This process is meant to decouple the envelope from the pitch in the time domain by extracting the slow-varying time components, which most likely correspond to the timbre and which will become the lowest coefficients, from the fast-varying time components, which most likely correspond to the pitch and which will become the highest coefficients [5].

A computational example

The discriminative power of the CQHCs can be measured by evaluating them for a simple instrument similarity task on the NSynth dataset [4], a large-scale and high-quality data set of annotated musical notes, which is publicly available at https://magenta.tensorflow.org/datasets/nsynth. The NSynth data set is composed of 305,979 musical notes generated from 1,006 instruments as 4 s, monophonic audio signals at a sampling rate of 16 kHz with pitches ranging over all the note numbers of a standard MIDI piano (21–108), and with five different velocities (25, 50, 75, 100, and 127), whenever applicable. The instruments are organized into 11 families, those being bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth_lead, and vocal, and three sources, namely, acoustic, electronic, and synthetic. The CQHCs are evaluated on the notes with a velocity of 75 only, leading to a subset of 60,388 different notes for 945 different instruments.

The CQHCs for all the notes in this subset were derived by first computing the CQT spectrogram using librosa, with a sampling rate of 16 kHz, a hop length of 512 samples, a minimum frequency of 32.70 Hz (corresponding to C1), 95 frequency bins, and 12 bins per octave, and then extracting 20 coefficients from the spectral component resulting from the deconvolution of the CQT spectrogram, leading to CQHCs of the size of 20 (coefficients) by 126 (time frames). A power CQT spectrogram was used and the real part of the spectral component was taken to ensure real values. For comparison, the MFCCs were also computed using librosa, with a sampling rate of 16 kHz, a window length of 1024 samples, a hop length of 512 samples, and 20 coefficients, leading to MFCCs the same size as the CQHCs, i.e., 20 by 126.

The cosine similarity for every pair of CQHCs and every pair of MFCCs (without repetition) was then computed after flattening the features into vectors of the length of 2,520 (20 by 126). These note similarities were averaged over every one of the 945 instruments, leading to similarity matrices of the size 945 by 945 for both the CQHCs and the MFCCs. Figure 3 presents these instrument similarity matrices for the CQHCs and MFCCs. As shown, the similarities for the CQHCs have more variance, while the similarities for the MFCCs are mostly very high (close to 1), showing poor discriminative power between the different instruments. Figure 4 additionally depicts self-similarities, i.e., the diagonal values in the similarity matrix (in green) and the error bars for cross-similarities, i.e., the off-diagonal values in the similarity matrix (means in red and standard deviations in yellow) for every instrument, for the CQHCs and MFCCs. As shown, the self-similarities for the CQHCs are noticeably higher than the means of the cross-similarities for most of the instruments, and generally higher than the means plus standard deviations as well, showing good discriminative power, while the self-similarities and cross-similarities for the MFCCs are all very high (close to 1). Also computed were the average self-similarities and cross-similarity means and standard deviations over all the 945 instruments for the CQHCs, giving 0.770, 0.513, and 0.208, respectively, and for the MFCCs, giving 0.939, 0.914, and 0.036, respectively.

The note similarities over every one of the 11 instrument families were also averaged, leading to similarity matrices of the size 11 by 11 for both the CQHCs

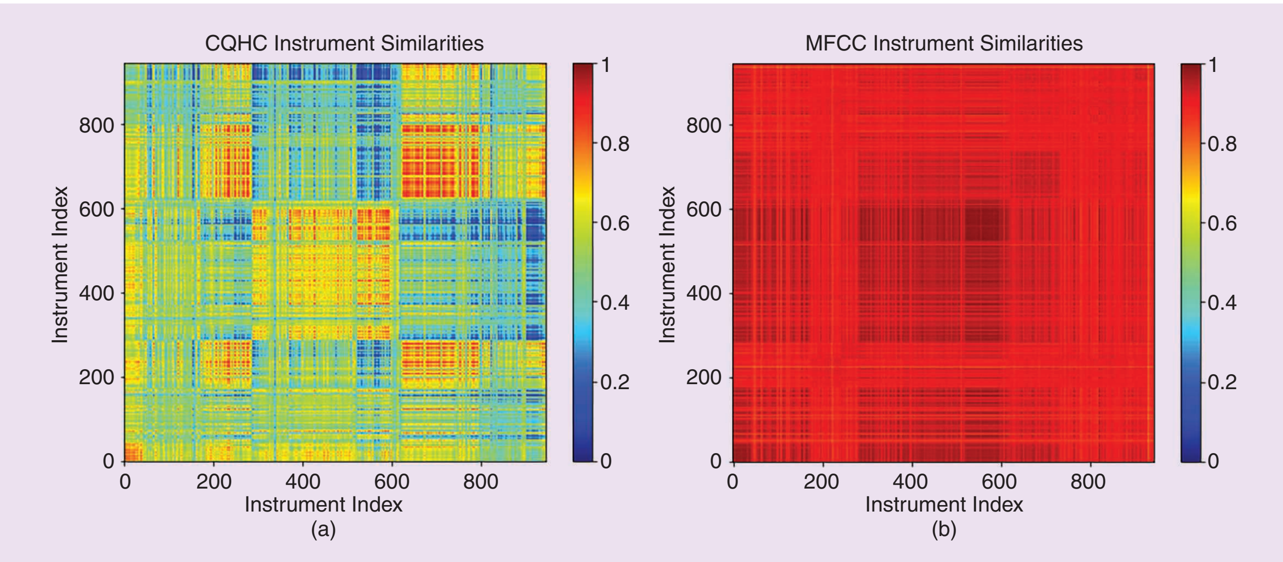

FIGURE 3. The similarity matrices computed for each pair of notes and averaged over every instrument of the NSynth subset for (a) the CQHCs and (b) the MFCCs.

and the MFCCs. Figure 5 depicts these instrument-family similarity matrices, and Figure 6 additionally shows the self-similarities (in green) and cross-similarities (means and standard deviations in red) for every instrument family for the CQHCs and MFCCs. As shown once again, the similarities for the CQHCs have more variance compared with the MFCCs, and their self-similarities are noticeably and mostly higher than the means plus standard deviations of their cross-similarities, while the self-similarities and cross-similarities for the MFCCs are all very high and harder to tell apart. We can clearly see with the CQHCs, for example, that the organ family has high self-similarity, showing that most of the instruments and notes in that family are quite similar to each other in terms of timbre. We can also see that the organ family is somewhat similar to the brass, flute, reed, and vocal families but quite different from the other families. On the other hand, the string family has fairly low self-similarity, showing that either the instruments or notes in that family differ substantially from each other, or that the CQHCs were not able to properly capture the overall timbre for this family.

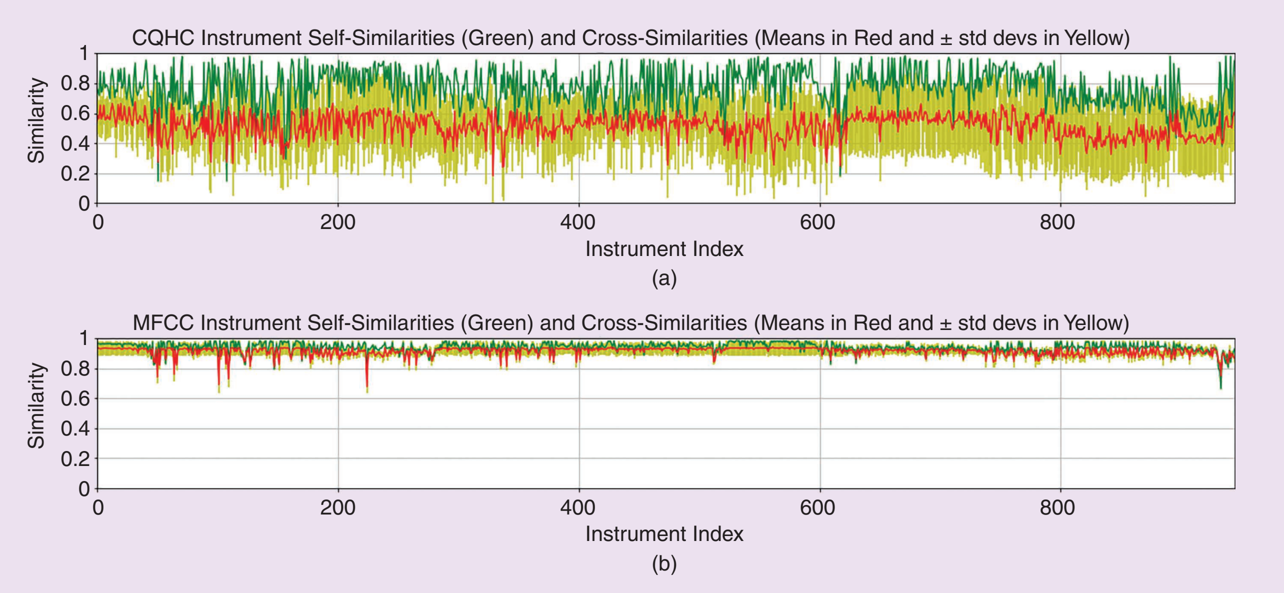

FIGURE 4. The self-similarities and cross-similarities derived from the instrument similarity matrices shown in Figure 3 for (a) the CQHCs and (b) the MFCCs. (std devs: standard deviations.)

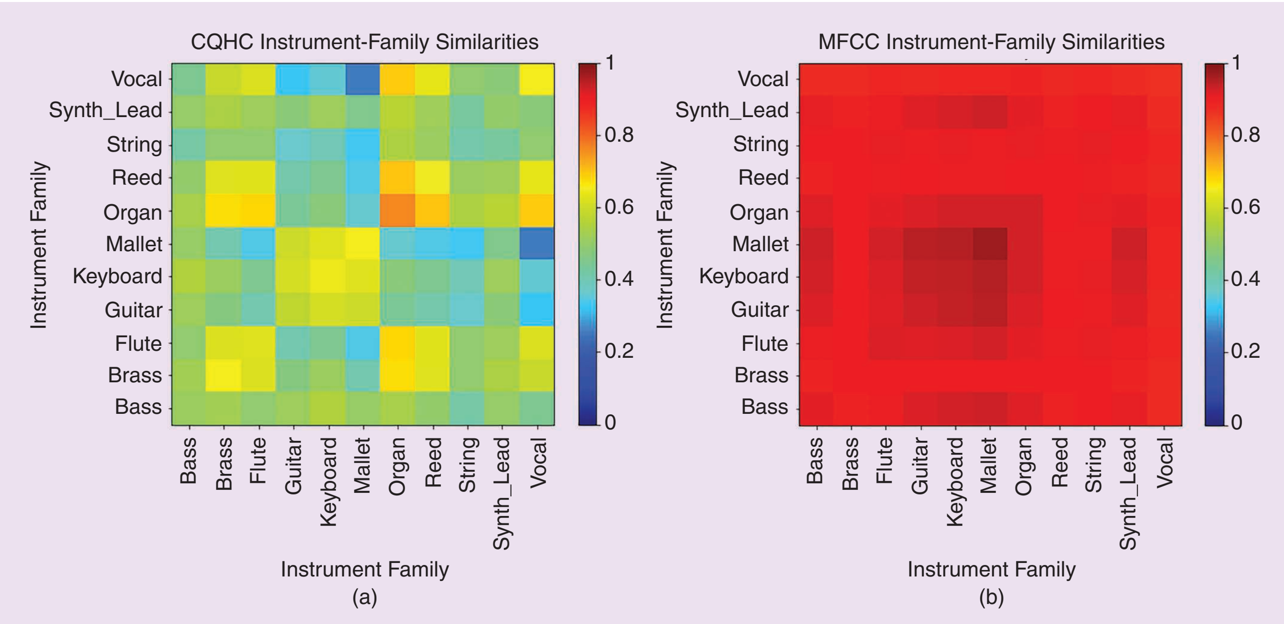

FIGURE 5. The similarity matrices computed for each pair of notes and averaged over every instrument family of the NSynth subset for (a) the CQHCs and (b) the MFCCs.

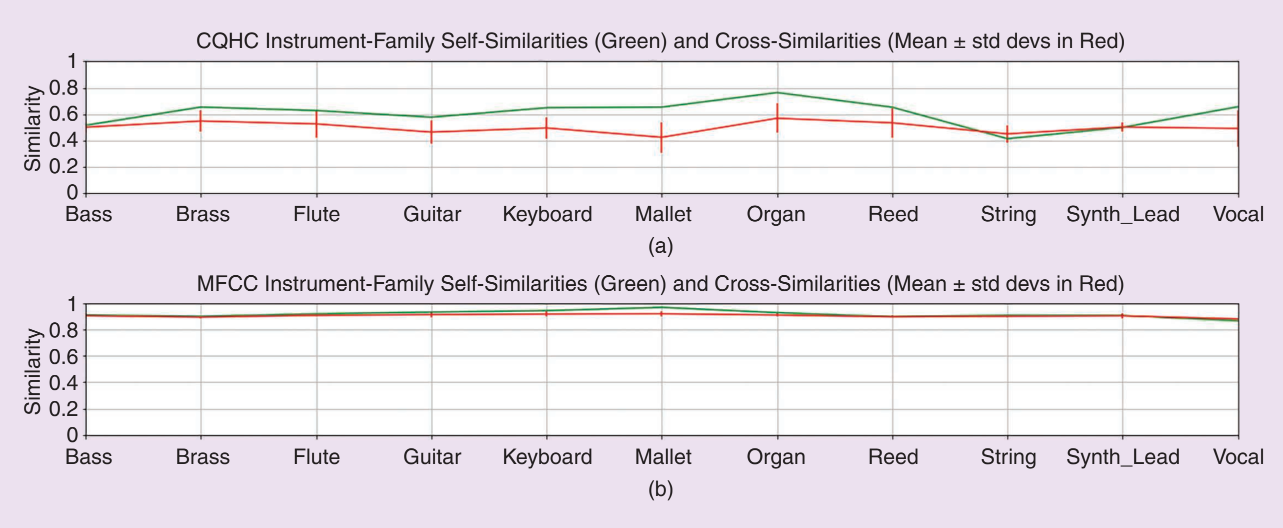

FIGURE 6. The self-similarities and cross-similarities derived from the instrument-family similarity matrices shown in Figure 5 for (a) the CQHCs and (b) the MFCCs.

We also computed the average self-similarities and cross-similarity means and standard deviations over all the 11 instrument families for the CQHCs, giving 0.605, 0.500, and 0.088, respectively, and for the MFCCs, giving 0.917, 0.905, and 0.013, respectively.

What we have learned

This article demonstrated that a simple but effective pitch-independent timbre feature well adapted to musical data could be designed by decomposing the CQT spectrum into a pitch-normalized spectral component and an energy-normalized pitch component and extracting a number of harmonic coefficients from the spectral component. These CQHCs provide a compact and interpretable feature for characterizing the timbre of a musical instrument, showing higher variance in terms of similarities on a simple instrument similarity task compared with the MFCCs, which are still often used by MIR practitioners to characterize timbre in music. For the interested reader, a Python implementation of the CQHCs online is provided with some examples at https://github.com/zafarrafii/CQHC-Python.

Author

Zafar Rafii (zafarrafii@gmail.com) received his Ph.D. degree in electrical engineering and computer science from Northwestern University in 2014. He is currently a research engineer manager at Gracenote, Emeryville, California, 94608, USA. His research interests are centered on audio analysis, including source separation and audio identification.

References

[1] B. C. J. Moore, *An Introduction to the Psychology of Hearing.* San Francisco, CA, USA: Academic, 2004.

[2] J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991, doi: 10.1121/1.400476.

[3] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoust. Soc. Amer.*, vol. 92, no. 5, pp. 2698–2701, 1992, doi: 10.1121/1.404385.

[4] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. 34th Int. Conf. Machine Learning*, Sydney, NSW, Australia, Aug. 6–11, 2017, pp. 1068–1077.

[5] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," in *Pattern Recognition and Artificial Intelligence*, San Francisco, CA, USA: Academic, 1976, pp. 374–388.

[6] M. Müller, *Information Retrieval for Music and Motion*. Berlin, Germany: Springer-Verlag, 2007.

[7] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *J. Acoust. Soc. Amer.*, vol. 61, no. 5, pp. 1270–1277, May 1977, doi: 10.1121/1.381428.

[8] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signals," *J. Acoust. Soc. Amer.*, vol. 130, no. 5, pp. 2902–2916, May 2011, doi: 10.1121/1.3642604.

[9] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 185–190, 1937, doi: 10.1121/1.1915893.

[10] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *Proc. 25th European Signal Process. Conf.*, Kos, Greece, Aug. 28–Sep. 2, 2017, pp. 2813–2817.

[11] Y.-N. Hung, I.-T. Chiang, Y.-A. Chen, and Y.-H. Yang, "Musical composition style transfer via disentangled timbre representations," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 10–16, 2019, pp. 4697–4703.

[12] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using Gaussian mixture variational autoencoders," in *Proc. 20th Conf. Int. Soc. Music Inf. Retrieval*, Delft, The Netherlands, Nov. 4–8, 2019, pp. 746–753.

[13] J. Lee, H.-S. Choi, J. Koo, and K. Lee, "Disentangling timbre and singing style with multi-singer singing synthesis system," in *Proc. 45th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 4–8, 2020.

[14] G. Fant, *Acoustic Theory of Speech Production*. The Hague, Paris: Mouton De Gruyter, 1970.

[15] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[16] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, Austin, TX, USA, Jul. 6–12, 2015, pp. 18–24, doi: 10.25080/Majora-7b98e3ed-003.

[17] C. Schoerkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. 7th Sound and Music Comput. Conf.*, Barcelona, Spain, Jul. 21–24, 2010.

[18] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, 1976, doi: 10.1109/TASSP.1976.1162830.

[19] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama, "Specmurt analysis of polyphonic music signals," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 16, no. 3, pp. 639–650, 2008, doi: 10.1109/TASL.2007.912998.

[20] B. P. Bogert, M. J. R. Healry, and J. W. Tukey, "The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe-cracking," in *Proc. Symp. Time Series Anal.*, 1963, pp. 209–243.