WAVELET SCATTERING FOR AUTOMATIC CHORD ESTIMATION

First Author
Affiliation1
author1@ismir.edu

Second Author
Retain these fake authors in
submission to preserve the formatting

Third Author
Affiliation3
author3@ismir.edu

ABSTRACT

State-of-the-art automatic chord recognition systems rely on multi-band chroma representations, Gaussian Mixture Model pattern matching, and Viterbi decoding. This paper explores the use of Haar wavelet transforms and scattering in place of multi-band chroma. Wavelets operating across octaves encode sums and differences in chroma bins at different scales. We describe both the Haar wavelet transform and deep wavelet scattering and develop an efficient algorithm for their computation. Potential benefits of wavelet representations, including stability to octave deformations, over multi-band chroma are discussed. Accuracy of wavelet representations used for chord recognition is analyzed over a large vocabulary of chord qualities.

1. INTRODUCTION

Along with lyrics and melody, chord sequences provide a succinct description of tonal music. As such, they are often written down under the form of lead sheets, for the use of accompanists and improvisers. Besides its historical purpose in music education and transmission, the knowledge of harmonic content has been leveraged in music information research to address higher-level tasks, including cover song identification [4], genre recognition [10], and lyrics-to-audio alignment [8].

A system for automatic chord estimation (ACE) typically consists of two stages: feature extraction and acoustic modeling. At the first stage, the audio query is converted into a time series of pitch class profiles, which represent the relative salience of pitch classes according to the twelve-tone equal temperament. At the second stage, each frame in the time series is assigned a chord label among a fixed vocabulary. We refer to the review of McVicar et al. [9] for a recent state of the art.

Chord estimation systems developed at NYU's Music and Audio Research Lab (such as those detailed in [2] [5]) use supervised learning approaches to train a chord model on a large chord vocabulary of 157 different chords. Typically, these systems consist of three-part pipeline:

Feature Extraction to transform the audio file into a musically relevant representation

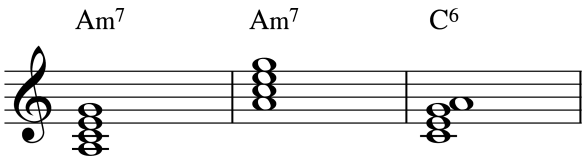

Figure 1. Three chords.

Pattern Matching to match the harmonic content in a particular time frame with a template in a trained chord model

Decoding to determine whether or not a predicted chord makes sense in context

Feature extraction in the ACE systems of [2] [1] uses a Constant-Q transform to create a frequency-space representation on a perceptual log-scale of ten octaves. That is, a time-frequency representation $\boldsymbol{X}[t, \gamma]$ encodes the energy in a temporal window t at a frequency band whose center is proportional to $2^{\gamma/12}$, where each γ corresponds to a pitch in twelve-tone equal temperament. A K-multi-band chroma representation is then created by first windowing the frequency scale with K Gaussians distributed evenly over the support of λ, and then by wrapping each windowed representation $\boldsymbol{X}[t, \gamma]$ into a chroma vector. This creates a new representation $\boldsymbol{Y}[t, q, k]$ which is invariant to octave transpositions where q is an integer $[\![0, 12[\![$ representing a chroma (or pitch) value, and K is the windowed band.

Pattern matching is done by creating a chord model for all 157 chords with Gaussian Mixture Models (GMMs). This is a supervised learning approach where features are extracted for a large training set that contains examples of all 157 chords (with annotations) and are then used to train GMM templates for each chord. Data augmentation is employed by rotating each chord through the chroma space so that one example of C minor can also train a model for C# minor, D minor, etc. For any temporal frame t, each of the K chroma representations $\boldsymbol{Y}_{t,q,k}$ are matched using GMMs and then fused together via geometric mean. Finally, **decoding** uses Hidden Markov Model (HMM) systems to post-filter the pattern matched data. Here, we use the Viterbi algorithm, which describes the transition probability from one chord C_i state to another – $\mathbb{P}(C_i \rightarrow C_j)$, therefore filtering out chord transitions that are highly unlikely. The matrix of transition probabilities for Viterbi decoding is generated from the labeled training data.

Our experiments looked to replace parts of the existing

feature extraction techniques with new ones based on Haar wavelet transforms and deep Haar scattering. We began with the MATLAB system designed by Taemin Cho [3] and added the new wavelet-based feature extraction routines to it.

2. EXPERIMENTAL SETUP AND EVALUATION

Evaluation of an ACE system was determined using some of the tools found in the mir_eval library [11] and written into a simple python script which compared the estimated chords coming out of the ACE system to the ground truth annotations for all songs in the testing set. All experiments were run on the High Performance Computing (HPC) environment at NYU's Courant Institute.

For each experiment, a chord model and Viterbi transition probability matrix are generated from a training set of 451 songs. The 'band' (K) of any experiment determines the amount of chroma vectors at any given temporal window – equivalent to the number of bands in the multiband chroma representation, and the maximum wavelet scale in the wavelet and scattering representations (i.e. the number of wavelet coefficients). Chord recognition is then carried out on the testing set of 65 songs. Both the training set and testing set of songs are kept constant across all experiments.

After generating estimated chord labels for each song in the test set, a Python script evaluates the results. As per [11], there is "no single right way to compare two sequences of chord labels," and the mir_eval package computes ACE accuracy based along metrics such as root, triads, maj/min, sevenths, inversions, etc. For simplicity, we evaluate all experiments using the mirex measure, which "considers a chord correct if it shares at least three pitch classes in common" [11].

State of the art systems such as those in [2] use a multiband chroma feature extraction with $K = 4$ windowed bands. To begin, some baseline trials were run on a training set consisting of 108 songs from the Beatles discography, 99 RWC pop songs, 224 songs from the Billboard dataset, and 20 Queen songs, for a total of 451 songs. Our testing dataset comprised of 65 songs from the Beatles and uspop datasets that were not part of the training set and that contained a sufficient number of examples of each chord quality.

The constant-Q spectrogram $\boldsymbol{X}[t, \gamma]$ is reshaped into a time-chroma-octave representation $\boldsymbol{X}[t, q, u]$ where the chroma index $q \in [\![0; Q[\![$ and octave index $u \in [\![0; K[\![$ are derived from the log-frequency index γ by Euclidean division: $\gamma = Q \times u + q$. All subsequent operations apply to the octave variable u, and are vectorized in terms of time t and chroma u. To alleviate notations, we replace the three-way tensor $\boldsymbol{X}[t, q, u]$ by a vector $\boldsymbol{x}[u]$, thus leaving the indices t and q implicit.

3. HAAR WAVELET TRANSFORM

Dating back to 1909, the Haar wavelet ψ is a piecewise constant, real function of compact support, consisting of two steps of equal length and opposite values. Within a discrete framework, it is defined by the following formula:

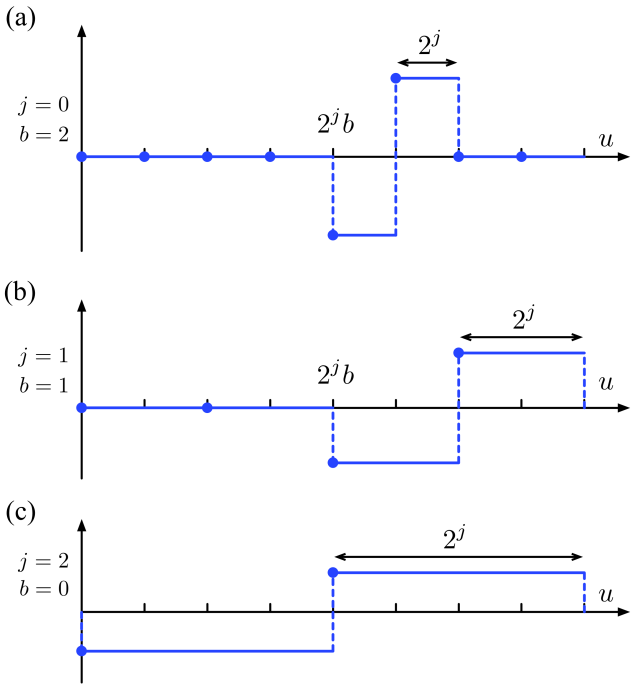

Figure 2. Three elements of the Haar wavelet basis $\{\boldsymbol{\psi_{j,b}}\}$ for various values of the scale index j and the translation index b. See text for details.

$$\forall u \in \mathbb{Z},\ \boldsymbol{\psi}[u] = \begin{cases} \frac{-1}{\sqrt{2}} & \text{if } u = 0 \\ \frac{1}{\sqrt{2}} & \text{if } u = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The "mother" wavelet $\boldsymbol{\psi}[u]$ is translated and dilated by powers of two, so as to produce a family of discrete sequences $\boldsymbol{\psi_{j,b}}[u] = 2^{j/2}\boldsymbol{\psi}[2^j(u - 2b)]$ indexed by the scale parameter $j \in \mathbb{N}$ and the translation parameter $b \in \mathbb{Z}$. After endowing them with the Euclidean inner product

$$\langle \boldsymbol{\psi_{j,b}} | \boldsymbol{\psi_{j',b'}} \rangle = \sum_{u=-\infty}^{+\infty} \boldsymbol{\psi_{j,b}}[u] \boldsymbol{\psi_{j',b'}}[u], \tag{2}$$

the wavelets $\{\boldsymbol{\psi_{j,b}}\}_{j,b}$ form an orthonormal basis of finite-energy real sequences. Moreover, the Haar wavelet is the shortest function of compact support such that the family $\{\boldsymbol{\psi_{j,b}}\}_{j,b}$ satisfies this orthonormality property. On the flip side, it has a poor localization in the Fourier domain, owing to its sharp discontinuities. Elements of the Haar wavelet basis are shown on Figure 3 for various values of j and b.

The wavelet transform coefficients of some finite-energy sequence $\boldsymbol{x} \in \ell^2(\mathbb{Z})$ are defined by $\boldsymbol{W}\boldsymbol{x}[j, b] = \langle \boldsymbol{x} | \boldsymbol{\psi_{j,b}} \rangle$. Since $\boldsymbol{x}[u]$ has a finite length $K = 2^J$, the above decomposition is informative only for indices (j, b) such that $j < J$ and $2^j b < K$, that is $b < 2^{J-j}$. The number of coefficients in the Haar wavelet transform of $\boldsymbol{x}[u]$ is thus equal to $\sum_{j<J} 2^{J-j} = 2^J - 1$. For the wavelet representation to preserve energy and allow signal reconstruction,

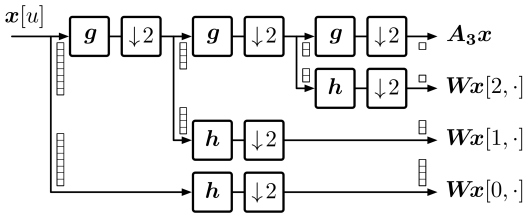

Figure 3. Discrete wavelet transform of a signal of length 8, as implemented with a convolutional pyramid scheme. See text for details.

a residual term

$$\boldsymbol{A_J x} = \boldsymbol{x}[0] - \sum_{j,b} \langle \boldsymbol{x} | \boldsymbol{\psi_{j,b}} \rangle \boldsymbol{\psi_{j,b}}[0] = \sum_{u<K} \boldsymbol{x}[u] \quad (3)$$

must be appended to the wavelet coefficients. Observe that $\boldsymbol{A_J x}$ computes a delocalized average of all signal coefficients, which can equivalently be formulated as an inner product with the constant function $\boldsymbol{\phi}[u] = 2^{-J/2}$ over the support $[\![0; K[\![$. Henceforth, it corresponds to the traditional chroma representation, where spectrogram bands of the same pitch class q are summed across all K octaves.

Since the wavelet representation amounts to K inner products in $\mathbb{R}^K$, its computational complexity is $\Theta(K^2)$ if implemented as a matrix-vector product. Implementing these inner products as convolutions would bring the complexity to $\Theta(K(\log_2 K)^2)$ by using the Fast Fourier Transform (FFT) algorithm. To improve this, Mallat has developed a recursive scheme, called *multiresolution pyramid* [7], which operates as a cascade of convolutions with some pair of quadrature mirror filters $(\boldsymbol{g}, \boldsymbol{h})$ and progressive downsamplings by a factor of two.

$$\boldsymbol{W x}[j, b] = (\boldsymbol{h_{\downarrow 2}} \circ \boldsymbol{g_{\downarrow 2}} \circ \ldots \circ \boldsymbol{g_{\downarrow 2}} \boldsymbol{x})\,[b] \quad (4)$$

$$\boldsymbol{A_J x} = \boldsymbol{g_{\downarrow 2}} \circ \ldots \circ \boldsymbol{g_{\downarrow 2}} \boldsymbol{x} \quad (5)$$

The associated Z-transforms G and H of $\boldsymbol{g}$ and $\boldsymbol{h}$ satisfy the quadrature mirror property $G(z) = H(-z)$ and energy conservation: $|G(z)|^2 + |H(z)|^2 = 1$. The interested reader can refer to chapter 7 of Mallat's textbook [6], of which we adopt the mathematical notations.

In the case of Haar wavelets, the low-pass filtering $(\boldsymbol{x} * \boldsymbol{g})$ consists of the sum between adjacent coefficients, whereas the high-pass filtering $(\boldsymbol{x} * \boldsymbol{h})$ is the coresponding difference, up to a renormalization constant:

$$(\boldsymbol{x} * \boldsymbol{g})[2b] = \frac{\boldsymbol{x}[2b+1] + \boldsymbol{x}[2b]}{\sqrt{2}} = \boldsymbol{x}[2b] - \langle \boldsymbol{x} | \boldsymbol{\psi_{0,b}} \rangle \quad (6)$$

$$(\boldsymbol{x} * \boldsymbol{h})[2b] = \frac{\boldsymbol{x}[2b+1] - \boldsymbol{x}[2b]}{\sqrt{2}} = \langle \boldsymbol{x} | \boldsymbol{\psi_{0,b}} \rangle. \quad (7)$$

When applied to the chroma representation $\boldsymbol{X}(t, q, z)$, where t is a temporal window, q a chroma pitch class (an integer between 1 and 12), and z an octave, the Haar wavelet transform encodes the sums and differences in chroma bins across octaves. The number of output "bands" from the Haar wavelet transform, given by k, determines the number of scales J for which the basis Haar wavelet is dilated: $2^J = k$.

	operations	memory
Matrix-vector product	$\Theta(K^2)$	$\Theta(K)$
Fast Fourier transform	$\Theta(K(\log K)^2)$	$\Theta(K)$
Multiresolution pyramid	$\Theta(K \log K)$	$\Theta(K)$

Table 1. Computational complexity and memory usage of various implementations of the Haar wavelet transform, for a one-dimensional signal of length K. See text for details.

The use of the Haar wavelet transform in the feature extraction stage for our ACE system is motivated by the fact that it is stable to octave deformations – something that the multiband representation is not. By coding for chroma relationships across varying octave scales, we are able to capture representations at both low scales (less invariance, more stability) and higher scales (higher invariance, less stability), thereby reducing the variance of the GMMs.

Upon receiving a CQT-derived pitch representation of 8 octaves $\boldsymbol{X}[t, \lambda]$, the lowest and highest octaves are thrown out of a total of 8 octaves (this follows suit from the multiband chroma representation). $\boldsymbol{X}$ is therefore a matrix of size $T \times \Lambda$, i.e. the number of frames in the song T times the perceptual frequency bins $\Lambda = (12 \times 8) = 96$. A chroma representation is then created by reshaping $\boldsymbol{X}[t, \lambda]$ into a tensor of size $T \times Q \times Z$ where $Q = 12$ (number of chroma pitch classes) and $Z = k = 8$ (number of octaves as well as number of wavelet coefficients). In the sequel, we keep the notation $\boldsymbol{X}[t, q, z]$ for simplicity.

The Haar wavelet transform is then computed iteratively over the scale $J = log_2(k)$ and stored into a cell of size k, with each row in the cell containing a chroma matrix of size $T \times Q$ (same as in the multiband case, but with $k = 8$ cell rows instead of 4). At each scale $j \in [\![1, J]\!]$, the chroma tensor $\boldsymbol{X}[t, q, z]$ is reshaped to a four dimensional tensor g from which the sum g and difference h coefficients are calculated:

$$\boldsymbol{X}[t, q, z] \rightarrow g[t, q, w, v] \quad (8)$$

where

$$\text{size}(\boldsymbol{X}) = (T \times Q \times Z) \quad \rightarrow \quad \text{size}(g) = (T \times Q \times 2 \times \frac{k}{2^j}) \quad (9)$$

h takes the difference of g along the third dimension ($\partial g / \partial w = h$). For each $v \in [1, \frac{k}{2^j}]$, the modulus $|h_{t,q,w,v}|$ is stored as a row in the output Haar wavelet representation cell. g is then summed along the third dimension (w) and squeezed down into a three dimensional tensor (singleton dimensions due to sum removed). After iterating at all scales, the resulting g is stored as the last (k'th) row in the output Haar cell array.

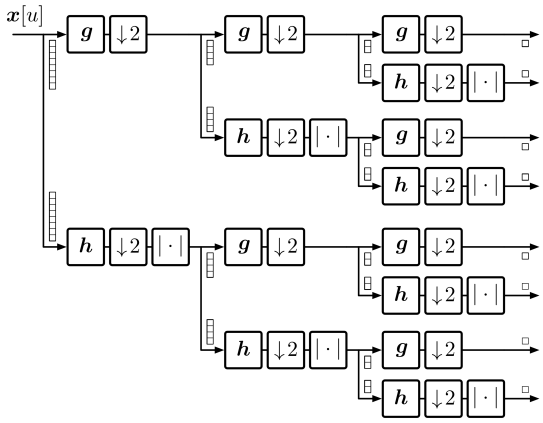

Figure 4. Deep scattering transform of a signal of length 8, as implemented with a convolutional pyramid scheme. See text for details.

	operations	memory
Matrix-vector product	$\Theta(K^3)$	$\Theta(K^2)$
Fast Fourier transform	$\Theta(K^2(\log K)^2)$	$\Theta(K^2)$
Multiresolution pyramid	$\Theta(K^2 \log K)$	$\Theta(K^2)$

Table 2. Computational complexity and memory usage of various implementations of the deep Haar scattering transform, for a one-dimensional signal of length K. See text for details.

4. DEEP HAAR SCATTERING

Two neighboring samples $\boldsymbol{x}[2b]$ and $\boldsymbol{x}[2b+1]$ can be retrieved, up to a permutation, from the absolute values of the subsampled QMF decomposition $(\boldsymbol{g}_{\downarrow \mathbf{2}}\boldsymbol{x}, \boldsymbol{h}_{\downarrow \mathbf{2}}\boldsymbol{x})$:

$$\max_{u \in \{2b, 2b+1\}} \boldsymbol{x}[u] = \frac{|\boldsymbol{g}_{\downarrow \mathbf{2}}\boldsymbol{x}|[b] + |\boldsymbol{h}_{\downarrow \mathbf{2}}\boldsymbol{x}|[b]}{\sqrt{2}} \tag{10}$$

$$\min_{u \in \{2b, 2b+1\}} \boldsymbol{x}[u] = \frac{|\boldsymbol{g}_{\downarrow \mathbf{2}}\boldsymbol{x}|[b] - |\boldsymbol{h}_{\downarrow \mathbf{2}}\boldsymbol{x}|[b]}{\sqrt{2}} \tag{11}$$

As opposed to the Haar wavelet transform representation, a Haar scattering representation concatenates sum and difference terms at different scales and then iterates. From the chroma representation $\chi_{t,q,z}$, we reshape to a tensor of dimension $J+2$ called ϕ:

$$\chi_{t,q,z} \to \phi_{t,q,\sigma_1,\sigma_2,\ldots,\sigma_J} \tag{12}$$

where the size of ϕ is $T \times Q \times 2_1 \times 2_2 \ldots \times 2_J$. At each scale $j \in [1, J]$, we first create a copy of ϕ called $\hat{\phi}$, and then concatenate the sum of $\hat{\phi}$ along scale σ_j and the difference of $\hat{\phi}$ along σ_j, setting the result as ϕ. We then take the modulus $|\phi|$ and iterate.

5. RESULTS

Results and discussion forthcoming.

6. DISCUSSION

Results and discussion forthcoming.

7. REFERENCES

[1] Taemin Cho. *Improved Techniques For Automatic Chord Recognition From Music Audio Signals*. PhD thesis, New York University, 2013.

[2] Taemin Cho and Juan P. Bello. On the relative importance of individual components of chord recognition systems. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(2):477–492, February 2014.

[3] Taemin Cho, Juan P. Bello, and Eric J. Humphrey. tmc-ace-mirex2013. `https://github.com/marl/tmc-ace-mirex2013`, 2013.

[4] Daniel P. W. Ellis and Graham E Poliner. Identifying "cover songs" with chroma features and dynamic programming beat tracking. In *Proc. ICASSP*, 2007.

[5] Eric J. Humphrey and Juan P. Bello. Four timely insights on automatic chord estimation. In *ISMIR*, 2015.

[6] Stéphane Mallat. *A Wavelet Tour of Signal Processing, 3rd edition: The Sparse Way*. Academic press, 2008.

[7] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.

[8] M. Mauch, H. Fujihara, and M. Goto. Integrating additional chord information into hmm-based lyrics-to-audio alignment. *IEEE TASLP*, 20(1):200–210, Jan 2012.

[9] M. McVicar, R. Santos-Rodrguez, Y. Ni, and T. D. Bie. Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):556–575, Feb 2014.

[10] Carlos Pérez-Sancho, David Rizo, and José M Inesta. Genre classification using chords and stochastic language models. *Connection science*, 21(2-3):145–159, 2009.

[11] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, 2014.