A COMPARISON OF PITCH CHROMA EXTRACTION ALGORITHMS

Miguel Perez[♯♭], Holger Kirchhoff[♯], Xavier Serra[♭]
Huawei Technologies Munich Research Center[♯]
Music Technology Group, Universitat Pompeu Fabra[♭]
miguel.perez.fernandez@huawei.com
holger.kirchhoff@huawei.com
xavier.serra@upf.edu

ABSTRACT

The pitch chroma is a popular way to represent pitch information in an octave independent way, with applications in automatic chord recognition, cover song identification, audio-to-score alignment, and others. Early chroma extraction algorithms employed expert knowledge to derive pitch chromas from short-time spectra. With the rise of deep learning, the emphasis moved from algorithm design to the structure of the network and the selection of appropriate training data. The approaches perform differently for various types of audio input. We conducted a set of experiments in order to explore the qualitative properties that each algorithm exhibits. These include how the number of concurrent pitches influences the chroma representation, and how noise or unpitched percussion can degrade the performance of the algorithms. We performed a quantitative analysis of various algorithms under these scenarios. The results show that chromas based on deep learning show huge potential, especially when it comes to noise reduction and ignoring non-tonal aspects of the music. However, we also found that some deep learning based chromas fail to accurately detect pitches at lower polyphony levels. We reflect on these results and discuss some paths to improvements for future chroma extraction algorithms.

1. INTRODUCTION

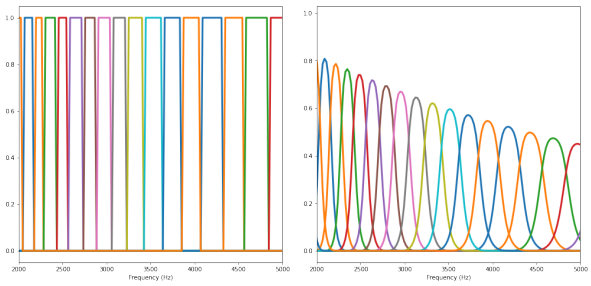
Data representation is an important aspect in designing systems that analyze and process real-world data. In the case of natural language processing it is common to use the words themselves with special tokens to indicate the beginning and end of phrases. In computer vision, the RGB channels of the images are used to feed the algorithms. In both of these areas, an unprocessed representation of the data can be used as the input for the systems. For music data, approaches that directly act on time-domain sample data have been less common. This can be attributed to the fact that the waveform representation is inherently difficult to interpret for humans and hence makes it difficult to design expert systems without any intermediate representation. More practically, machine learning systems that act on waveform data turned out to be significantly harder to train and usually result in larger model sizes that require more training data [1]. Intermediate representations that transform the raw audio into some more interpretable representation, however, have proven useful even for machine learning systems, as they can reduce the dimensionality of the input data and often lead to more accurate and robust results. Since music can be understood in different aspects, such as rhythm, melody, instrumentation, genre, etc., different intermediate representations such as STFT spectrograms, CQT spectrograms or Mel spectrograms, or features derived from these have been extensively used for analysis algorithms.

The pitch chroma is a feature addressing the tonal information contained in a music signal. It is sometimes denoted as pitch-class profile (PCP) or simply as chroma. It encodes tonal information of music in an octave independent representation, also known as pitch-class. A chroma is a vector of usually 12 dimensions, each representing the presence of a pitch class (C, C#, D, . . . , B). Finer pitch resolutions like quarter tones (a chroma with 24 dimensions) can be used, but are less common in the literature. Chromas are usually extracted for short, consecutive blocks of audio, resulting in a *chromagram* representation for an extended section of music. Pitch chroma representations can be considered more robust than those accounting for octave height, since the compression to a single octave eliminates wrong pitch estimates in a different octave (octave errors) caused by ambiguous harmonic patterns in the spectrum. Although initially designed for automatic chord estimation (ACE), chromas became useful in a wide range of tasks, such as cover version identification (CVI) [2], audio-to-score alignment [3] and music creation [4].

Given the above definition and applications for pitch chromas, a chroma extraction algorithm is expected to fulfil a number of tasks when transforming audio to chromas. Generally, a chroma extractor should eliminate any type of irrelevant spectral content in order not to obscure the tonal information. Irrelevant content can be any transient, non-tonal components (percussive or noise-like) such as drums, cheering audience, etc. We here neglect the fact that percussive instruments might also have a tonal component to them. In the same way, overtones should not contribute to pitch classes other than their corresponding fundamental frequency. It is arguable whether a chroma should represent the tonal content *as it is*, or whether it should incorporate further processing steps depending on the target

application, e.g temporal smoothing in the case of ACE. In any case, a transparent processing structure would require a chroma to only represent those pitch classes that are present at each point in time. Subsequent processing stages would decide then, if a set of notes belong to an arpeggiated chord, or if some notes are ornaments that do not belong to the chord itself.

The rest of the paper is divided into four sections. In Section 2 we give an overview of the most popular algorithms, highlighting the main differences and evolution over time. In Section 3 we describe the experiments we have carried out in order to do a qualitative analysis of the different approaches. The results derived from the different experiments are presented in Section 4. Section 5 contains our reflections on the properties of the different chromas and possible future research directions.



(a) Fujishima's [5] filters (b) Ellis & Poliner filterbank [6]

Figure 1: Filterbanks in the 2-5 kHz range. Each color represents a different pitch class. Fujishima rectangular filters make abrupt changes between pitch assignments. Ellis & Poliner penalize the deviations from the ideal frequencies.

2. RELATED WORK

In this section, we present an overview of various notable approaches to chroma extraction. We divided the algorithms into three different classes: knowledge-based chromas, deep chromas from chord labels, and multipitch deep chromas. In the first category, we included chroma algorithms that only use expert knowledge and digital signal processing techniques to obtain chroma representations. The second one contains chroma algorithms that make use of deep learning with ACE datasets. Lastly, we introduce chroma algorithms trained with datasets containing pitch annotations instead of chord labels.

2.1 Knowledge-based Chromas

Chroma algorithms evolved since the first algorithm presented by Fujishima [5]. In his work, he presents a bank of non-overlapping rectangular filters (see Figure 1a) that maps STFT magnitude spectra to pitch classes. After applying the filters to the spectrum, the energy of all filter outputs belonging to the same pitch class are accumulated to form the pitch chroma. The fact that the filters are non-overlapping makes them quite selective w.r.t the frequency components. The rectangular shape of the filters has the effect that all frequency components in a semitone range contribute equally to the pitch class. Since the tonal components of interest will usually appear closer the the center of the filter, it might be beneficial to give less weight to components that deviate from the center, in order to reduce the influence of non-tonal (noise, percussion) and spurious components. Following this idea, Ellis & Poliner [6] proposed a filterbank of Gaussian filters with a semitone spacing. With this shape, frequency components further away from the center of the filter will contribute less to that specific pitch class. Also, to reduce the influence of percussion or other elements, the filters emphasize mid-frequencies, where most of the tonal content will be located. See Figure 1 for a comparison of Fujishima and Ellis & Poliner filters.

With both these approaches, energies of harmonics which contribute to the pitch perception of their fundamental are here erroneously assigned to different pitch classes. E.g. the 3rd harmonic is a perfect fifth above the fundamental and will hence not contribute to the pitch class of its fundamental, but to the pitch class a fifth above. This leads to false contributions to the resulting pitch chroma. Gómez [7] therefore proposes to extract *harmonic* pitch-class profiles (HPCP). This method differs from the previous methods in two significant ways: Firstly, it only considers peaks of the spectrum instead of complete frequency bands, and it secondly also maps harmonics to the pitch class of their fundamental frequency. The use of peaks aims to reduce the influence of undesired elements, such as unpitched percussion or background noise. The mapping of harmonics is achieved by accumulating the energies not only of the fundamentals but also of their overtones with decreasing weight. The contribution of the harmonics to the fundamental can be seen as a pattern-matching mechanism.

The NNLS chroma proposed by Mauch & Dixon [8] employs another pattern matching mechanism to identify harmonic structures in the short-time spectrogram. Their system first maps STFT magnitude spectra onto a log-frequency axis with a 1/3-semitone resolution. Given a dictionary of prototypical spectra for a pitch range range from A0 to G#6, approximate note activations are extracted by means of a non-negative least squares (NNLS) algorithm. The note activations are then summarized to form the pitch classes for each instantaneous chroma.

Another popular chroma representation with pattern matching mechanisms is the ***c**hroma DCT-**r**educed log **p**itch* (CRP) [9]. This approach is inspired by mel-frequency cepstral coefficients (MFCCs), a popular representation for speech, which produces a set of coefficients where the firsts ones are closely related to timbre [10]. During the MFCC extraction process, the discrete cosine transform (DCT) captures periodicities present in the spectrum. After mapping the STFT magnitude spectra onto a log-frequency axis with a semitone resolution, the authors apply the DCT as it is done during the MFCC extraction process to extract a number of coefficients. The information concerning timbre is discarded by setting the first n coefficients to zero, and the inverse DCT is expected to return a chroma with improved robustness to timbre. Both the number of coefficients to

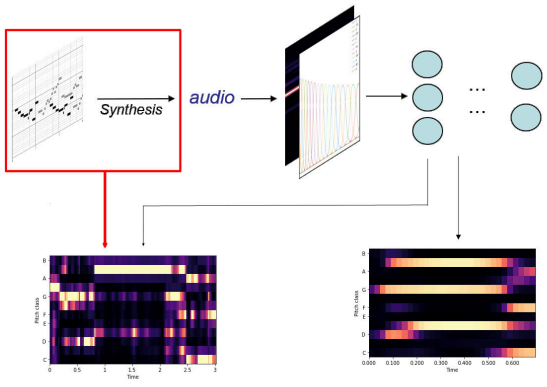


Figure 2: The general schema to train deep chromas. The audio signal is converted to an intermediate representation such (e.g: CQT), which is used as input for the neural network that returns the chromagrams. The red box illustrates the case of Wu & Li [11] where a MIDI score is used to create the audio and later to serve as the ground-truth

extract and to discard are parameters that must be set according to the use case.

2.2 Deep Chromas: Chord Labels

Pattern matching mechanisms help to distinguish between fundamentals and harmonics, but also add even more parameters in order to adapt to multiple situations: The tuning reference, the number of harmonics and peaks to consider, the parameters used to create the *note dictionary* in NNLS chromas, the number of MFCCs to retain, etc. Recalculating the optimal parameters for different music styles, instrumentation and possible sonorities become impractical for many applications. With the arrival of deep learning (DL), researchers switched from manually designing these algorithms and tuning their parameters to letting neural networks (NN) learn them from examples. We call this set of chroma extraction algorithms *deep chromas*.

Most of the chromas extracted using these techniques were designed to improve the accuracy of ACE systems, mostly because a critical amount of data was available for western pop music, making it possible to train deep learning systems. In their work, Korzeniowski & Widmer [12] designed a system named the *deep chroma extractor* (DCE) that learns to extract chromas from the output of a filterbank. This is to the best of our knowledge the very first deep chroma, and its architecture is based one of the earliest kinds of NNs, the multilayer perceptron. Given a set of audio signals and corresponding chord labels, specific chroma targets are set up that contain activations of those pitch classes that correspond to the annotated chord at that time instance. For the estimation of a single chroma instance, the authors consider a context window of approximately 0.7 seconds around the time instance. This provides the neural network the opportunity to take preceding and subsequent content into account when estimating each chroma.

Another algorithm that extracts chroma features from audio for the purpose of ACE was proposed in [13]. Instead of using a fully connected layer and context frames, the authors employed convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The CNN part learns convolution kernels to convolve with the input spectrogram, looking for patterns related to pitch sensations. RNNs are a particular type of neural network that can consider information about previous and posterior audio frames. Here, the RNN part learns to ignore spurious changes in the spectrum through time, eliminating the necessity of using fixed-length context frames as in the *deep chroma extractor*.

2.3 Deep Chromas: Multi-pitch Labels

Since the deep chroma approaches of the previous section are trained on chord-based chroma targets, those systems will usually output a set of concurrent pitches. This might be acceptable or even desired for ACE applications, however, it makes those extractors less suitable for applications that rely on the analysis of more detailed pitch-class information. To obtain a more accurate representation, instead of using chord labels, Wu & Li [11] employ chroma targets based on note annotations, i.e. onset time, offset time and pitch. This enables the network to capture only the active pitch classes at each point in time. This network is based on a CNN. It uses the harmonic constant Q transform (HCQT) [14] as its input representation, which associates each CQT bin with its corresponding harmonics. This allows the network to see fundamental frequencies and harmonics simultaneously. Due to the lack of sufficient real-world data with corresponding note annotations, the authors revert to synthesized MIDI data. 6000 MIDI files were collected from the RWC Classical, Jazz and Genre dataset [15], and the Lakh MIDI dataset [16] ensuring a diverse range of musical styles. These files were synthesized using a sample-based SoundFont to create the audio input of the network. The problem of creating an audio dataset of real instruments with finer pitch-class information is that it would require note annotations including pitch, onset and offset times with sufficient precision. This laborious process does not scale easily to the amount of required data for training deep NNs.

Weiss et al. [17, 18] proposed to circumvent this problem by using real audio recordings of classical music together with non-aligned MIDI scores. To temporally match input and output data of the network, the authors employ the connectionist-temporal-classification (CTC) loss. The CTC was originally proposed for automatic speech recognition [19] where ground truth sentences require temporal alignment with speech utterances. This loss does not require a precise alignment between audio and score. Instead it only relies on the correct order of note events in both the MIDI score and the audio.

3. EXPERIMENTS

The chroma extractor families introduced in the previous section follow different design principles and partly serve

Polyphony	1	2	3	4	5	6
# Examples	1798	723	963	955	390	205

Table 1: Number of examples per polyphony level.

different purposes. We are interested in analyzing the behaviour of these systems for different types of audio input. More specifically, the accuracy of these systems was studied for audio inputs with varying levels of polyphony, as well as their ability to suppress non-tonal elements. For this purpose, we conducted a number of experiments in which we measured these properties quantitatively. A number of relevant use cases was defined and corresponding pairs of audio/chromagram were set up for each of those. To obtain audio/ground-truth chromagram pairs, we use a synthesized MIDI dataset with individual instrument stems (see Subsection 3.4). The audio of those pairs were then processed by a selection of existing chroma extraction algorithms. The output of each system was compared with the target chromas and metrics were computed to measure the similarity between actual and target chromas.

3.1 Varying Polyphony

In a first set of experiments, we evaluated the accuracy of the chroma algorithms for different levels of polyphony, i.e. different numbers of concurrent pitches. For that purpose, the dataset was divided into sections of 1, 2, 3, ... up to 6 concurrent pitches. We ensured that each fragment was at least 4 seconds long, resulting in several hundred examples for each polyphony level (see Table 1). We expect the chromas to contain high values at the present pitch classes and low values at all others. The accuracy of the systems is measured as the cosine similarity of each chromagram output $\mathbf{o}$ and the corresponding target chroma $\mathbf{t}$:

$$S(\mathbf{o}, \mathbf{t}) = \frac{\mathbf{o}^T \cdot \mathbf{t}}{||\mathbf{o}|| \cdot ||\mathbf{t}||} \tag{1}$$

These similarities cover a range from 0 to 1, with 0 indicating complete dissimilarity and 1 identical chromas. We call the cosine similarity between the algorithm's chroma and the ground truth *chroma accuracy*.

3.2 Suppression of Non-tonal Elements

A second set of experiments looked at the suppression of non-tonal components in the chroma output. As discussed in Section 1, chromas are expected to only capture the tonal content. Any non-tonal components should not contribute to the result. To measure the influence of percussive elements, we selected fragments from the datasets containing percussion with a minimum length of 10s. The individual stems of the dataset allowed us to store a version of each fragment without percussive elements alongside a version containing the full mix. The target chromas are the same for both cases since the percussive elements do not contribute to the targets. Again, we measure the cosine similarity between the actual and the target chromas. Percussive components are very common, particularly in popular music tracks, however, those elements usually have limited durations and hence only affect a part of the spectrogram. Stationary noise, on the other hand, poses a different challenge, covering a much wider time-frequency range and is also present in otherwise silent sections. Therefore we also evaluated the chroma extractors on the input sequences with added white noise instead of percussion. The ratio of tonal and non-tonal components depends on the mix and is most likely not fixed across music tracks. We hence evaluated percussion and noise at various levels of intensity. We use the signal-to-noise ratio (SNR) to characterize how present they are in relation to the tonal elements. Given a signal x with the tonal elements, and the signal y with only non-tonal elements, we define our SNR as:

$$SNR_{dB} = 10\, log_{10} \left(\frac{\sum_i x_i^2}{\sum_j y_j^2} \right) \tag{2}$$

In this case, we compare the algorithm's output for the signals x and y. Note that by varying the intensity of the non-tonal elements, we do not want to compare how much the algorithm resembles the ground truth, but how much of these elements' presence is able to achieve a significant change at the resulting chromagram. Instead, as we compare how similar is the resulting chromagram with and without the presence of non-tonal elements, we call in this case the result of the cosine distance *chroma similarity*.

Algorithm	Type	Name
Ellis & Poliner	Knowledge-based	Ellis
HPCP	Knowledge-based	Gomez
NNLS	Knowledge-based	Mauch
DCE	Deep: chords	Korzeniowski
McFee & Bello	Deep: chords	McFee
Wu & Li	Deep: multi-pitch	Wu
Weiss & Peeters	Deep: multi-pitch	Weiss

Table 2: The algorithms used in our experiments, along with the type of algorithms. The name column indicates the way we will refer to these algorithms in the results.

3.3 Tested Systems

We selected seven existing chroma extraction systems. An overview can be found in Table 2.

From the knowledge-based chromas (see Section 2.1) we employed Ellis & Poliner's system [6] as implemented in Librosa [20] [1] . For HPCP by Gómez we used the Essentia library [21]. In the case of NNLS we used the original vamp plugin [2] with a python wrapper; note that this algorithm returns a chroma for bass frequencies and another one for treble frequencies. We only used the treble chroma output of this algorithm.

For the family of chromas based on chord labels (see Section 2.2), we selected two algorithms. The DCE by Korzeniowski & Widmer as implemented in the Madmom library [22], which partially differs from the original model but according to the authors achieves similar results. For

[1] https://librosa.org/doc/main/generated/librosa.feature.chroma_stft.html

[2] http://www.isophonics.net/nnls-chroma

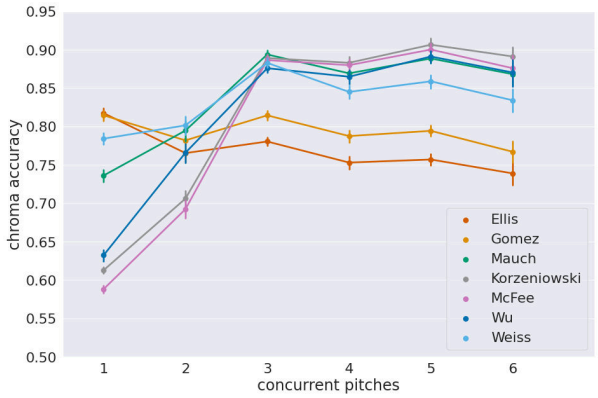


Figure 3: Mean chroma accuracy for each algorithm according to the number of pitches present. The algorithms that perform better for smaller levers of polyphony (Ellis and Gomez) perform worse at higher levels and viceversa.

the algorithm by McFee & Bello [13], we used the the public implementation provided by the authors [3].

For the family of deep chromas based on multi-pitch labels (see Section 2.3) we selected two algorithms: Wu & Li [11], and Weiss et al [18] with the public implementations given by the authors [4] [5]. Note that from the two works previously mentioned by Weiss, we selected the one using prealignment since it seems to provide slightly better results.

Each of the chromas based on deep learning operate at their own hop size, FFT size, and sample rate. For all the knowledge-based algorithms we used a hop size of 4410, a FFT size of 8096, and a sample rate of 44100.

3.4 Dataset

For all experiments, the Slakh dataset [23] was employed, which contains audio synthesized from 2100 files in the MIDI Lakh dataset (140 hours in total). There are 34 different instrument categories that cover a wide range of musical instrument timbres. The MIDI files were used to set up our target chroma representations for each track. The target chromas were sampled with the same hop size as the algorithms.

4. RESULTS

4.1 Varying Polyphony

The results for the polyphony experiments can be seen in Figure 3. This figure shows the mean chroma accuracy for the different levels of polyphony described in section 3.1.

We can observe two main tendencies: Chroma algorithms with less intricate pattern matching mechanisms such as Ellis or Gomez, perform better at lower levels of polyphony, but achieve lower chroma accuracies as the number of concurrent pitches grows. The rest of the algorithms on the other hand, perform worse for 1 or 2 concurrent pitches,

[3] https://github.com/bmcfee/crema

[4] https://github.com/Xiao-Ming/ChordRecognitionMIDITrainedExtractor

[5] https://github.com/christofw/pitchclass_mctc

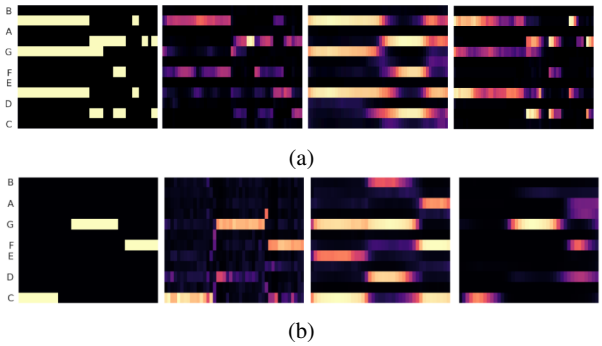


Figure 4: From left to right, the target chromas and chromagrams from Gomez, Korzeniowski, and Weiss.

but then reach a high chroma accuracies for 3 or more concurrent pitch-classes.

Deep chromas based on chords are at the lower part of this chart for 1 and 2 concurrent pitches. This is an expected behaviour since the models were trained on chord labels that contain three or more pitches, making the outputs of these algorithms usually chord-like chromas.

Deep chromas trained on multi-pitch labels follow the same trend as those trained on chord chromas: they exhibit better results for higher polyphony. But while the algorithm of Weiss is almost as good as Ellis or Gomez for 1 and 2 pitches, the chroma accuracy of Wu is significantly lower. We hypothesize that this could be attributed to the datasets used to train the models. Wu was trained with synthesised MIDI files from various datasets which consist mostly of pop music. This results in just a few passages where there are just one or two notes being played simultaneously. In contrast, Weiss was trained with classical music, which is more likely to have solo passages or sections where several instruments playing in unison.

To provide some intuition for these result, Figures 4a and 4b show chromagrams for two example sequences. The first is an excerpt of multiple concurrent pitches. The algorithm from Gomez gets some of the pitches right but struggles to clearly show all simultaneous notes; The DCE by Korzeniowski shows concurrent pitches more clearly, but at the same time smoothes the activations over time, resulting in 'chord-like' activations. The algorithm by Weiss generally shows less clear activations than the DCE but overall captures the finer details of the target chromagrams while at the same time recognising concurrent pitches. The second example in Fig. 4b shows a short melody with only a single active pitch at each time instance. Gomez' algorithm clearly highlights the correct pitch classes, but also contains spurious peaks, most likely caused by overtones that were not correctly assigned to their fundamentals. The DCE by Korzeniowski on the other hand contains activations for pitches that are actually not present in the audio at all. In order to produce chord-like chromas, the additional pitches form a major triad above the actual pitch. The extraction algorithm by Weiss only contains spurious activations for the last note.

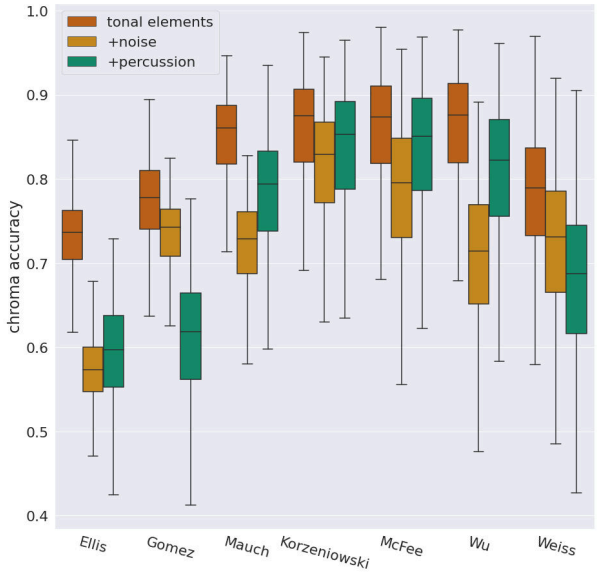


Figure 5: Chroma accuracy in 3 different scenarios: Just tonal information, added noise, and percussion. The algorithms performed best when only tonal elements were present in the signal. After adding noise or percussion ($SNR_{dB} = -15$) the output of the chromagrams resembled less to the targets.

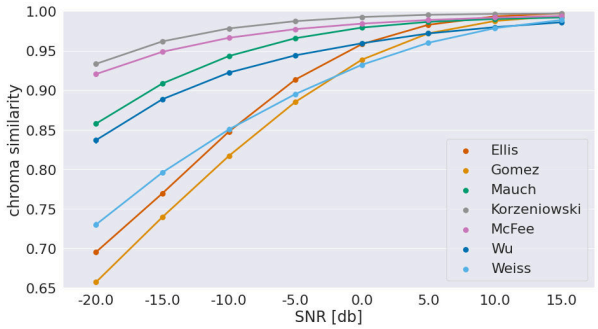


Figure 6: Chroma similarity as a function of percussion intensity. A smaller slope means that the algorithm is less affected by the presence of percussion.

4.2 Suppression of Non-tonal Elements

Figure 5 shows the results of the experiments with non-tonal components. It can clearly be seen that the addition of noise and percussion in all cases degrades the accuracy of the chromas. While the algorithms by Korzeniowski, McFee and Wu are least affected by the presence of non-tonal elements, all other algorithms show significant losses in accuracy. However, noise and percussion do not affect the algorithms in the same way. The peak selection from Gomez for example deals very well with noise, however, it struggles to suppress percussive elements in the spectrum. The opposite is the case for the algorithms by Mauch and Wu: noise seems to affect these algorithms more than percussion.

Figures 6 and 7 show how the intensity of the non-tonal elements (percussion and noise, respectively) affects the

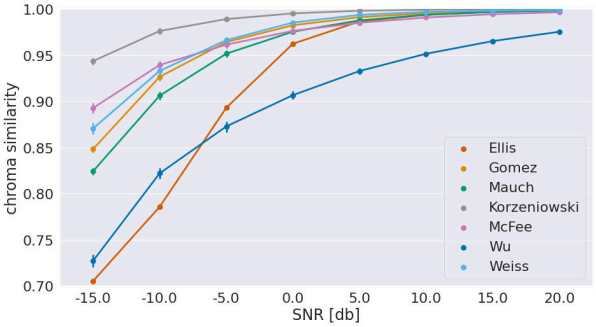


Figure 7: Chroma similarity as a function of noise intensity. A smaller slope means that the algorithm is less affected by the presence of noise.

algorithms. Some algorithms were able to deal with the non-tonal elements even with an $SNR_{dB} = -15$, and after $SNR_{dB} = 0$ most of the algorithms had a chroma similarity of 0.95 for both scenarios. Notice however that Wu is affected by noise even when this is quite small compared to the tonal signal.

5. CONCLUSIONS

In this work we investigated the performance of several existing pitch chroma extraction algorithms, analyzing their capability to deal with tonal content of increasing polyphony, as well as their robustness against non-tonal components. Given a dataset of music tracks with corresponding MIDI ground truth, pitch chromas were extracted by each algorithm for each scenario, and their accuracy w.r.t. a target chroma representation was measured. Results showed that chroma extraction algorithms based on deep learning produce a more accurate representation for higher polyphony levels and are generally more robust in suppressing non-tonal components. However, their reduced accuracy for lower polyphony levels hints at biases in the corresponding training sets.

Chromas are a well known and widespread feature in MIR. We argue that a generic chroma extraction algorithm should capture the tonal content as it is, thereby neither adding pitches that are not present in the audio nor performing additional processing steps such as temporal smoothing. While this might be obvious for applications such as audio-to-score alignment for which a temporal resolution at the note level is required [3], it will also be beneficial for ACE as it disentangles the detection of active pitch classes from the interpretation by a musical model.

Our results in Fig. 5 show that chroma extraction algorithms overall yield meaningful representations with decent accuracies. However, even for the case of music containing only tonal components, median accuracies do not exceed the 90% mark. With additional percussion or noise, these accuracies decrease. This shows that there is still room for improvement in the overall quality of the algorithms. The fact that the algorithm by Wu & Li achieves the highest results for content with only tonal components, encourages us to think that using multi-pitch content for the training of chroma algorithms is indeed worthwhile and

might pave the way to more accurate chroma representations.

A crucial factor for the training of better systems, however, is the choice of training data. Obviously, a sufficient amount of real-world audio with precise annotations is required, but also other qualitative data properties seem to be important: a diverse number of timbres and the presence of unpitched percussion and potentially non-musical sounds. Biases in the number of concurrent pitches should be addressed as well. This could be either by balancing the levels of polyphony in the training data or using loss functions that can diminish the number of false positives, such as the Weighted Binary Cross Entropy [24].

Acknowledgments

This research took place as part of a collaboration between the Huawei Munich Research center and the Pompeu Fabra University in Barcelona.

6. REFERENCES

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016.

[2] F. Yesiler, J. Serra, and E. Gomez, "Accurate and scalable version identification using musically-motivated embeddings," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, may 2020, pp. 21–25.

[3] C. Joder, S. Essid, and G. Richard, "A comparative study of tonal acoustic features for a symbolic level music-to-score alignment," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 409–412.

[4] G. Bernardes, M. Davies, and C. Guedes, "A hierarchical harmonic mixing method," in *Music Technology with Swing*. Springer International Publishing, 2018, pp. 151–170.

[5] T. Fujishima, "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music," in *International Computer Music Conference Proceedings*, vol. 9, no. 6, 1999, pp. 464–467.

[6] D. P. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, 2007, pp. IV–1429–IV–1432.

[7] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, University Pompeu Fabra, Barcelona, Spain, July 2006.

[8] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *International Society for Music Information Retrieval Conference*, 2010, pp. 135–140.

[9] M. Muller and S. Ewert, "Towards timbre-invariant audio features for harmony-based music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 649–662, mar 2010.

[10] H. Terasawa, M. Slaney, and J. Berger, "The thirteen colors of timbre," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* IEEE, 2005, pp. 323–326.

[11] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 355–366, feb 2019.

[12] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," *International Society for Music Information Retrieval Conference*, pp. 37–43, 2016.

[13] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition." in *International Society for Music Information Retrieval*. Zenodo, 2017.

[14] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music." in *International Society for Music Information Retrieval*. Zenodo, 2017.

[15] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical, and jazz music databases," in *International Conference on Music Information Retrieval*, 10 2002, pp. 287–288.

[16] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, Columbia University, 2016.

[17] C. Weiss, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, "Learning Pitch-Class Representations from Score- Audio Pairs of Classical Music," in *International Society for Music Information Retrieval Conference*. Online: International Society for Music Information Retrieval, Nov. 2021, pp. 746–753.

[18] C. Weiss and G. Peeters, "Training Deep Pitch-Class Representations With a Multi-Label CTC Loss," in *International Society for Music Information Retrieval Conference*. Online: International Society for Music Information Retrieval, Nov. 2021, pp. 754–761.

[19] C. Wigington, B. Price, and S. Cohen, "Multi-label connectionist temporal classification," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, sep 2019.

[20] B. McFee, C. Raffel, D. Liang, D. 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*, vol. 8. SciPy, 2015.

[21] D. Bogdanov, X. Serra, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, and J. Zapata, "ESSENTIA," in *Proceedings of the 21st ACM international conference on Multimedia - MM '13*. ACM Press, 2013.

[22] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM international conference on Multimedia*. Amsterdam, The Netherlands: ACM, oct 2016, pp. 1174–1178.

[23] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[24] J. Zeitler, "Extracting tonal features for music analysis using deep learning," Master's thesis, ASC Major Research Project, Friedrich-Alexander-University of Erlangen-Nuremberg, Erlangen, 2020.