A FEATURE SMOOTHING METHOD FOR CHORD RECOGNITION USING RECURRENCE PLOTS (FOR MIREX2011 SUBMISSIONS)

Taemin Cho and Juan P. Bello
Music and Audio Research Laboratory (MARL)
New York University, New York, USA
{tmc323,jpbello}@nyu.edu

0. MIREX 2011 SUBMISSION

This submission contains three different chord models

- CB1 : 24 Binary templates, thus no training needed.
- CB2 : 25 Pre-trained GMMs (15 Gaussians with diagonal covariance matrices) using 179 Beatles, 20 Queen, and 100 RWC pop songs.
- CB3 : for Train/Test tasks, 25 GMMs (15 Gaussians with diagonal covariance matrices) will be trained by training set during the evaluations.

CB3 shows the best result in Train/Test tasks (avoiding overfitting over the known MIREX test set). In addition, CB1 that doesn't need training shows better result than other systems of Train/Test tasks.

ABSTRACT

In this paper, we propose a feature smoothing technique for chord recognition tasks based on repeated patterns within a song. By only considering repeated segments of a song, our method can smooth the features without losing chord boundary information and fine details of the original feature. While a similar existing technique requires several hard decisions such as beat quantization and segmentation, our method uses a simple pragmatic approach based on recurrence plot to decide which repeated parts to include in the smoothing process. This approach uses a more formal definition of the repetition search and allows shorter ("chord-size") repeated segments to contribute to the feature improvement process. In our experiments, our method outperforms conventional and popular smoothing techniques (a moving average filter and a median filter). In particular, it shows a synergistic effect when used with the Viterbi decoder.

1. INTRODUCTION

The majority of state of the art chord recognition systems are based on frame-wise analysis of chroma features extracted from an input signal. The chord sequence is determined by a pattern matching process that measures the fit between a set of predefined chord models and each frame of the input chromagram. In order to precisely identify chord boundaries, the frame rate of the chroma features is typically faster than the rate of chord changes in music. However, this makes the chroma features sensitive to local transients and noise in the signal. A popular choice to cope with this problem is to pre-process the chromagram using either a low-pass filter or a median filter prior to the pattern matching process. Both filters blur out transients and noise in the signal by smoothing the features across neighboring frames. Another favored approach is using a Viterbi decoder that finds the most likely sequence of chords based on the chord-type probabilities estimated from the pattern matching process. By reducing the number of chord transitions using a relatively high self-transition probability (the probability of remaining in a chord), the Viterbi decoder can filter out spurious transitions caused by short bursts of noise.

In our previous work [4], we found that the combination of pre-filtering (either a moving average filter or a median filter) and post-filtering (the Viterbi decoder) does not yield a synergistic impact on performance, although many systems use the combination [2, 6]. This is because the effects of pre-filtering substantially overlap with those of post-filtering, i.e. they carry out essentially the same function in the sense of constraining sudden movements over a short series of local frames.

In this paper, we propose a feature smoothing technique based on an important aspect of music, repetition. By averaging repeated chroma patterns within a piece of music, our method attenuates unsystematic deviations and noise and reinforces harmonic information of chroma frames. This method is inspired by the one proposed by Mauch et al. [6]. In their approach, the information about the repetitive structure of songs is used to enhance chroma features for chord estimation. They use a conventional frame-by-frame self-similarity matrix generated from a beat-synchronous chromagram. From the matrix, they extract repeated chord progressions of equal length by examining all diagonal lines. The beat and bar information estimated from a song play a crucial role in their greedy algorithm to find repeated sections. The found segments are merged into larger segment types (e.g. verse and chorus) without overlapping. Their new features are then obtained by averaging chroma features from multiple occurrences of the same segment type.

Unlike Mauch et al., our method decides which repeated parts to include in the smoothing process by a simple thresholding operation using the technique of recurrence plots. As our method doesn't use beat and bar information, it avoid the errors in the initial feature analysis (e.g. onset detection or beat tracking), that propagate through the subsequent processing stages and may hurt overall performance [8]. In our method, repeated sections are not limited to a few large units (e.g. chorus or verse), but include smaller units such as chords. Thus, our method can generate new chroma features using relatively many repeated frames collected from all across the song. As the repeated frames are assumed to have the same harmonic content, the smoothing only occurs within the same chords, thus preserving boundary information. In our experiments, this smoothing method yields better results than the conventional methods in all cases including the combination with the Viterbi decoder.

The remainder of this paper is structured as follows. In Section 2, we provide a detailed description of our method. In Section 3, we describe the data and evaluation methodology used in our experiments. The results and discussions are provided in Section 4, and our conclusions and directions for future work are presented in Section 5.

2. APPROACH

The block diagram of our feature smoothing process is shown in Figure 1. First, the audio signal is segmented and transformed into chroma features. The chroma features are then projected into phase space using time-delay embedding prior to calculating the recurrence plot. The weight matrix is derived from the recurrence plot, and combined with the original chroma features as a coefficient set in the feature reconstruction process. To measure the performance of our method on various types of chroma features, we evaluate our method on conventional chroma features and one of their most recent variants, CRP features [7]. The following subsections discuss the details of the approach including the feature set and our methodology for generating and applying the weight matrix to construct new chroma features.

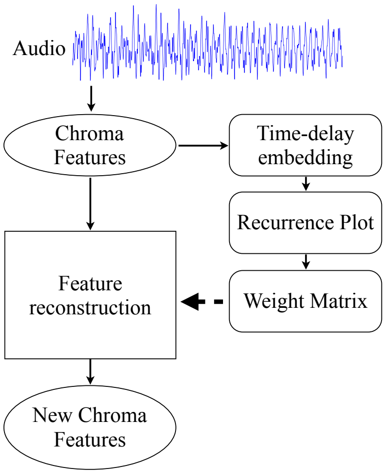


Figure 1. Block diagram of feature smoothing process based on recurrence plot.

2.1 Chroma Features

Pitch Class Profile (PCP), or chroma features, represent the energy of the audio signal present in each of the twelve pitch classes of the chromatic scale. In this paper, the chroma features are derived from a slightly modified version of the constant-Q transform [3] by mapping each frequency bin of the constant-Q spectrum to a corresponding pitch class. Let us define the k^{th} bin constant-Q kernel function as:

$$\mathcal{K}_k(m) = \omega_k(m) e^{-j2\pi f_k m}, \quad m \in [0, N_k - 1] \tag{1}$$

where ω_k is a Hamming window of length N_k, which varies with the center frequency f_k so that it has a fixed Q-value. The center frequency f_k is based on the equal tempered scale such that:

$$f_k = 2^{k/\beta} f_{\min} \tag{2}$$

where β is the number of bins per octave, and $f_{\min}$ is the minimum analysis frequency.

The constant-Q transform X_{cq} of a segmented audio signal $x(m)$, $m \in [0, N_{\text{seg}} - 1]$ is then calculated as:

$$X_{cq}(k) = \frac{1}{\min(N_{\text{seg}}, N_k)} \sum_{\nu=0}^{N-1} X(\nu) K_k^*(\nu) \tag{3}$$

where $N > N_k\ \forall k$, $X(\nu)$ is the N-point DFT of the signal, and $K_k^*(\nu)$ is the conjugate of the N-point DFT of the k^{th} kernel function. The signal and kernel functions are padded with trailing zeros to length N prior to applying the DFT. To prevent underestimation of low frequencies where $N_{\text{seg}} < N_k$, the smaller value between N_{seg} and N_k is used as the normalization factor. In this paper, we use $\beta = 36$, with the analysis performed between $f_{\min}$ = 27.5 Hz and $f_{\max}$ = 4186 Hz (i.e. corresponding to the MIDI pitches 21 to 108). The STFT window length N_{seg} is 8192 (186 ms), and hop size is 4096 (93 ms) samples at 44100 Hz sample rate.

A 12-bins per octave spectrum $P(p)$, $p \in [1, N_p]$ is obtained by combining adjacent bins of the $X_{cq}(k)$ using $\beta/12$-wide non-overlapping Gaussian windows. To avoid percussive noise (e.g. bass drums) in low frequencies and to attenuate the effect of non-harmonic tones caused by high-order harmonics in high frequencies, $P(p)$ is windowed with a Gaussian centered at C4 (MIDI pitch 60). Finally, a chroma vector $C = \{c_b\}$, $b \in [1, 12]$ can simply be calculated by folding the spectrum $P(p)$.

2.2 CRP Features

CRP (Chroma DCT-Reduced log Pitch) features, proposed by Müller et al. [7], are one of the most recent variants of conventional chroma features. Their derivation is inspired by Mel-frequency cepstral coefficients (MFCCs) which are popular in speech and music recognition. First, the spectrum $P(p)$ is logarithmized using $\log(P(p) \cdot \gamma + 1)$ with a suitable compression factor $\gamma > 1 \in \mathbb{R}$, and transformed to the *cepstral* domain using the Discrete Cosine Transform (DCT). The ξ-lowest coefficients of the resulting *cepstrum* are then set to zero. Finally, the cepstrum is transformed back using the inverse DCT, and the resulting pitch vectors are summarized into the 12-dimensional chroma vectors, C_{RP}. It is important to note that by removing the DC component from the cepstrum, a C_{RP} vector contains both positive and negative values. The feature vectors are then normalized by the ℓ^2-norm. In this paper, we use $\gamma = 1000$ and $\xi = 25$ as suggested by [7].

The main advantage of using CRP features for chord recognition comes from applying logarithmic compression on the spectrum $P(p)$. In conventional chroma features, melodies and bass lines are problematic, because they generate single high-energy peaks that dominate the chroma feature distributions to the detriment of the background harmony of the frame. The logarithm de-emphasizes the dominant pitch salience while boosting the background harmonic contents. In addition, by removing low coefficients from the cepstrum (i.e. formants, spectral shape), CRP features maximize the effect of compression and become invariant to changes in timbre.

2.3 Recurrence Plot and Weight Matrix

The weight matrix is computed using recurrence plot (RP) theory, which provides a sophisticated way to analyze sequential data [5], and have been previously used with chroma features in other MIR tasks such as cover version identification [9], and recently, in structural similarity [1]. A key feature of recurrence plots is the use of time-delay embedding. Time-delay embedding is a method for transforming a time series into a multidimensional sequence of lagged data. In other words, it provides a way to transform frame-by-frame analysis into n-gram analysis (i.e. subsequence-by-subsequence).

The n^{th} time-delay embedded chroma vector $\check{C}(n)$ can be constructed by concatenating all the elements of a chroma sequence $C(n) = \{c_b(n)\}$, $b \in [1, 12]$ from time n to $n + (M-1)\tau$ as:

$$\begin{aligned}\check{C}(n) = \big(&c_1(n), c_1(n+\tau), \ldots, c_1(n+(M-1)\tau), \ldots \\ &c_{12}(n), c_{12}(n+\tau), \ldots, c_{12}(n+(M-1)\tau)\big)\end{aligned} \tag{4}$$

$\check{C}(n)$ is then normalized to have unit length. The self-similarity matrix $S_{i,j}$ is calculated as:

$$S_{i,j} = \frac{\left|\left|\check{C}(i) - \check{C}(j)\right|\right|}{2} \tag{5}$$

where $i, j \in [1, N]$, N is the length of the time-delay embedded chroma sequence, and $||\cdot||$ is the Euclidean norm. The normalization factor (the constant 2 in the denominator) is the maximum possible distance value between unit length vectors. Hence, $0 \leq S_{i,j} \leq 1, \forall\, i, j \in [1, N]$.

Unlike a conventional frame-by-frame self-similarity matrix (i.e. a special case of $S_{i,j}$ with parameters $M = 1$ and $\tau = 1$), the additional embedding process makes the matrix more robust to short term noise or deviations by evaluating vectors of sample sequences (i.e. $M \cdot \tau$ length sequence) instead of using only samples. An RP can be obtained from $S_{i,j}$ with a suitable threshold ϵ as:

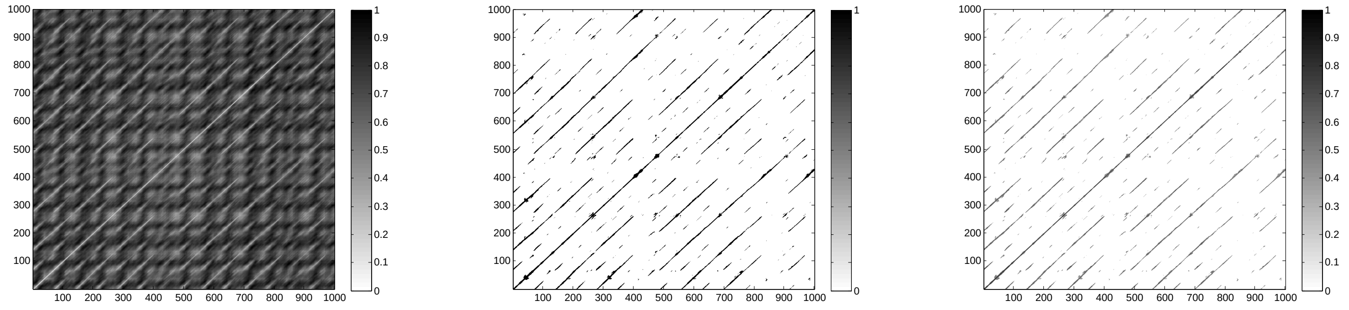
$$R_{i,j} = H(\epsilon - S_{i,j}), \quad i, j \in [1, N] \tag{6}$$

where H is the Heaviside step function. The choice of ϵ is important because it is the only criterion to determine which parts are actually repeated. However, a global thresholding with a fixed threshold is not appropriate in our case, because the useful range of thresholds can vary greatly between songs or even within a given song. A better strategy is to simply match the number of nearest neighbors in the phase space constructed by Eqn. (4). In this approach, $\epsilon(n)$ is defined as a threshold to ensure that $R_{i,n} = 1$ for the θ points closest to the n^{th} point of the trajectory. In practice, we expand this approach to both columns and rows of RP to include every possible repeated pattern in the smoothing process as:

$$R_{i,j} = H(\epsilon(n) - S_{i,n}) \vee H(\epsilon(n) - S_{n,j}) \tag{7}$$

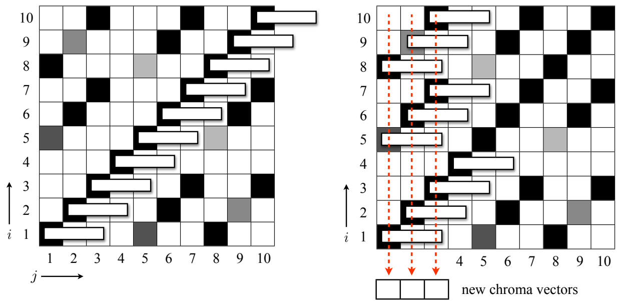
where $i, j, n \in [1, N]$. Finally, a weight matrix $W_{i,j}$ can be calculated using information from $S_{i,j}$ and $R_{i,j}$ as:

$$W_{i,j} = (1 - S_{i,j}) \cdot R_{i,j} \tag{8}$$



(a) $S_{i,j}$ (b) $R_{i,j}$ (c) $W_{i,j}$

Figure 2. (a) a similarity matrix ($M = 25, \tau = 1$), (b) a recurrence plot ($\theta = 50$), (c) a weight matrix



(a) $W_{i,j}$ (b) $W_{i,j}$

Figure 3. Reconstruction process: (a) overlapped chroma segments (white boxes), (b) chroma summation over overlapped segments with weight values.

Hence, the matrix is sparse and has real values indicating the similarity degrees of repeated sections. Figure 2 shows examples of a similarity matrix $S_{i,j}$, a recurrence plot $R_{i,j}$ and a weight matrix $W_{i,j}$ where $M = 25$, $\tau = 1$ and $\theta = 50$. In this paper, we fix $\tau = 1$ (i.e. no skipping frames).

2.4 Feature Reconstruction

Each column (or row) of $W_{i,j}$ contains information about recurrences of the current event across the whole song. More specifically, the i^{th} activated component (i.e. non-zero components) in the j^{th} column vector indicates that the i^{th} segment is similar to the j^{th} segment. For example, the first column of Figure 3(a) shows that the 5th and 8th segments are similar to the first segment. For $M = 3$ and $N = 10$, Figure 3(a) depicts the temporal validity of chroma vector M-grams.

To generate the first smoothed chroma vector from the example in Figure 3(a), the activated weights at $i = \{1, 5, 8\}$ of the first column are multiplied with the first frames of the corresponding segments. Then the results are summed up in the first smoothed chroma vector (see the left most down arrow in Figure 3(b)). Similarly, the second frame of the smoothed chromagram uses the weights on the second column and the first frames of the corresponding chroma segments (i.e. $i = \{2, 6, 9\}$). However, the overlapping means that the second frames from the previous segments should also be considered (see the second column in Figure 3(b)). More generally, the n^{th} frame of the smoothed chromagram is computed from the weights in the previous $n - M - 1$ columns. This process can be described as:

$$\hat{C}(n) = \sum_{m=0}^{M-1} \frac{\sum_{i=1}^{N} W_{i,n-m} \cdot C(i)}{\sum_{i=1}^{N} W_{i,n-m}} \qquad (9)$$

where the denominator is a normalization factor that adjusts for the contribution of overlapping chroma segments.

Figure 4(a) shows a chromagram and its smoothed versions using a moving average filter (Figure 4(b)), a median filter (Figure 4(c)) and our method (Figure 4(d)). The moving average filter used in Figure 4(b) is calculated as:

$$\bar{C}(n) = \frac{1}{\lambda} \sum_{d=0}^{\lambda-1} C\left(n + d - \left\lfloor \frac{\lambda-1}{2} \right\rfloor\right) \qquad (10)$$

and the median filter used in Figure 4(c) is defined as:

$$\begin{aligned} \tilde{C}(n) &= \underset{d}{\text{median}}\ C(d), \\ d \in \mathbb{N}, \quad n - \left\lfloor \frac{\lambda-1}{2} \right\rfloor &\leq d \leq n + \left\lceil \frac{\lambda-1}{2} \right\rceil \end{aligned} \qquad (11)$$

where λ is the number of adjacent frames to be processed. In Figure 4, the chromagram generated by our method is much cleaner than the original chromagram, while keeping sharp boundaries between chord segments. Figure 4(b), on the other hand, shows blurred boundaries, and the median filter in Figure 4(c) removes both the noise and the fine detail since it can't distinguish the difference between those signals.

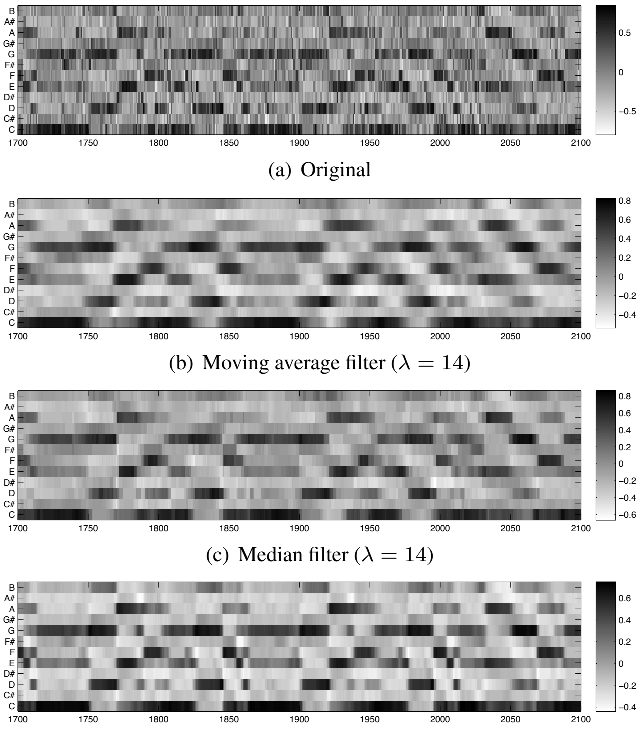
3. EVALUATION METHODOLOGY

The experiments are performed on 249 chord annotated songs. The data set comprises 179 songs [1] from Christopher Harte's

[1] "Revolution 9" from *The White Album* is removed from the experiment due to its lack of harmonic content.

	Without Viterbi Decoder				With Viterbi Decoder			
	None	Mean	Median	Our method	None	Mean	Median	Our method
C	49.93	65.51 (14)	66.22 (14)	69.23 (25, 47)	72.02	71.67 (4)	72.54 (4)	74.81 (25, 10)
C_{RP}	54.26	71.16 (14)	71.05 (14)	72.85 (25, 50)	75.36	75.76 (4)	75.64 (4)	**77.91** (25, 15)

Table 1. Average accuracies of the binary template model with no filtering (labeled ‘None’), a moving average filter (labeled ‘Mean’), a median filter, our method, and their combinations with the Viterbi decoder. The optimal parameters are given in parentheses, (λ) for both a moving average filter and a median filter, and (M, θ) for our method.



(a) Original

(b) Moving average filter ($\lambda = 14$)

(c) Median filter ($\lambda = 14$)

(d) Our method ($M = 25, \theta = 50$)

Figure 4. Chromagrams: (a) an original chromagram excerpt from “Let It Be” by The Beatles, (b) a smoothed chromagram using a moving average filter with $\lambda = 14$, (c) a median filter with $\lambda = 14$, and (d) our method with $M = 25, \theta = 50$.

Beatles dataset, 20 songs from Matthias Mauch’s Queen dataset and 50 pop songs from the RWC (Real World Computing) database manually annotated by music students at NYU. The evaluations are performed on 12 major, 12 minor triads and a no-chord detection task. In the evaluation, audio frames where the RMS is under -57 dB are assumed to be no-chords.

For the pattern matching process, binary chord templates and multivariate Gaussian Mixture Models (GMMs) are used. The binary chord templates (for 12 major and 12 minor triads) are manually generated based on basic chord theory. In a 12-dimensional binary chord template vector, each component corresponding to a chord-tone is set to 1, and the other components are set to 0 (e.g. [1 0 0 0 1 0 0 1 0 0 0 0] for a C Major triad, where the left to right order of the vector components follows the chromatic scale from C). The detected chord on one given frame is the one whose template is closest to the chroma vector of the frame in an Euclidean sense. The pseudo-probabilities for applying the Viterbi decoder are calculated by taking the reciprocal of the Euclidean distances.

The parameters of the multivariate GMMs are estimated from annotated training data using the EM algorithm. For training, the data is segmented based on the chord annotations and transposed to the C-based chord. The root-normalized chord collection is used to train C-major and C-minor models that are then re-transposed to the remaining roots to define the 22 models. In this paper, we use a mixture of 15 Gaussians with diagonal covariance matrices.

For the Viterbi decoder, the transition penalty ρ is applied. The transition penalty adjusts the strength of the self-transition probability relative to transitions between different chords [4]. It is applied as follows:

$$\log(\hat{a}_{i,j}) = \begin{cases} \log(a_{i,j}) - \rho & \text{for } i \neq j \\ \log(a_{i,j}) & \text{for } i = j \end{cases} \quad (12)$$

where $A = [a_{i,j}]$ is the original transition probability matrix and $\hat{A} = [\hat{a}_{i,j}]$ is the modified matrix with penalty ρ. For A, we use a uniform transition probability matrix in which all chord transitions have the same probability, hence $A_{i,j} = 1/24, \forall\, i, j \in [1, 24]$

For statistical models (GMMs), each experiment is performed using a 10-fold cross validation on 10 randomly classified groups; 9 groups contain 25 songs each, and one group contains 24 songs. For each iteration, one group is selected as a test set, and the remaining 9 groups are used for training. The chord recognition rate is calculated as follows:

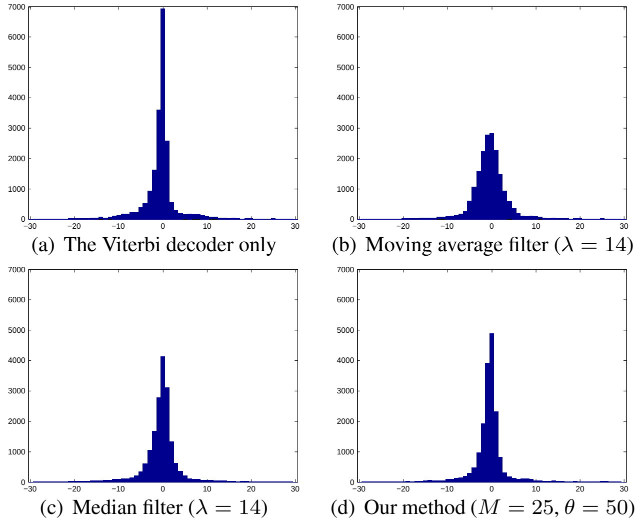
$$\text{Accuracy} = \frac{\text{total duration of correct chords}}{\text{total duration of dataset}} \times 100\% \quad (13)$$

4. RESULTS AND DISCUSSION

Table 1 shows the average accuracies of the binary template model with a moving average filter, a median filter, our method, and their combinations with the Viterbi decoder. The results show that C_{RP} yields better results than C in every case.

	None	Mean	Median	Our method
C	73.85	73.52 (3)	74.41 (4)	75.78 (25, 6)
C_{RP}	77.82	77.63 (4)	77.69 (4)	**79.61** (25, 9)

Table 2. Average accuracies of GMMs. The optimal parameters are given in parentheses, (λ) for both a moving average filter and a median filter, and (M, θ) for our method.



(a) The Viterbi decoder only (b) Moving average filter ($\lambda = 14$)

(c) Median filter ($\lambda = 14$) (d) Our method ($M = 25, \theta = 50$)

Figure 5. Distributions of deviations between annotated and detected boundaries: C_{RP} and binary template model with the Viterbi decoder: (a) The Viterbi decoder only, prefiltering with (b) a moving average filter, (c) a median filter, and (d) our method. In the graph, the X-axis means the distance between annotated and detected chord boundaries in frames, and the Y-axis means the number of boundaries belonging to the distances.

Also they show that our method outperforms the use of conventional filters regardless of the types of features. Table 2 shows the result of using GMMs with the different combinations of the filters. Similar to the case of the binary template model, C_{RP} performs better than C, and our method maintains its advantages against both a moving average filter and a median filter. All differences between conventional methods and our method are significant in paired t-test at $p < 0.01$.

One notable difference between our method and the conventional filters is its compatibility with the Viterbi decoder. As shown in both tables, unlike our method, the moving average filter has almost no impact on the overall performance when used in combination with the Viterbi decoder. This is due to the blurred boundaries caused by the filter, as seen in Figure 4(b). Figure 5 shows the distributions of deviations (in frames) between annotated and detected boundaries of the combinations of different pre-filtering methods and the Viterbi decoder. For our goal, a sharp and narrow distribution is ideal, since it means little deviation from the ground truth. In the case of Figure 5(d), the number of frames used to generate a new frame is at least 50 ($\theta = 50$). As shown in Figure 5(b), although the moving average filter employs a relatively small number of frames ($\lambda = 14$) for smoothing, it shows larger deviations than our method in Figure 5(d).

Although the median filter is much better at preserving sharp edges than the moving average filter as shown in Figure 5(c), the results in Table 1 and Table 2 are not much better than those of the moving average filter. In the case of C_{RP}, the median filter shows about the same performance as the moving average filter. The median filter is efficient at removing impulsive noise. However, in whitened feature space such as C_{RP}, it has little influence on the performance, but rather may lead to appreciable loss in signal details, because it uses only rank-order information of the input data within the filter window without considering its original temporal-order information. These characteristic errors of conventional filters hurt the performance. In the case of Figure 5(b), the accuracy rate is 72.2%, and for Figure 5(c), the accuracy rate is 72.7% for C_{RP} features (compared to 75.4% for Figure 5(a) and 76.4% for Figure 5(d)). On the contrary, since our method keeps deviations low and also preserves fine details, it maximizes the benefits of both our method and the Viterbi decoder.

5. CONCLUSION

In this paper, we provided a feature smoothing method based on repeated patterns. By applying recurrence plot theory, our method smoothes chroma features using information from harmonically-related frames from the whole sequence, as opposed to conventional smoothing where only a few adjacent frames are used. We showed that this method contributes to performance improvement by preserving the benefit of a fast-frame-rate analysis (i.e. sensing precise chord boundaries) while alleviating its problems (i.e. noise and transients). This advantage is maintained among different types of chroma features.

In our experiments, we applied the same parameters (M and θ) to all songs, despite the risk of over-smoothing. In the future, we plan to develop adaptive methods for optimally choosing these parameters for each individual track. We fully expect this adaptation to improve performance beyond what is reported in this paper.

6. REFERENCES

[1] J.P. Bello. Measuring structural similarity in music. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(7):2013 – 2025, 2011.

[2] J.P. Bello and J. Pickens. A robust mid-level represen-

tation for harmonic content in music signals. In *Proc. ISMIR*, pages 304–311, 2005.

[3] J.C. Brown and M.S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92:2698–2701, 1992.

[4] T. Cho, R.J. Weiss, and J.P. Bello. Exploring common variations in state of the art chord recognition systems. In *Proc. SMC*, 2010.

[5] N. Marwan, M. Carmen Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6):237–329, 2007.

[6] M. Mauch, K. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. ISMIR*, pages 231–236, 2009.

[7] Meinard Müller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.

[8] L. Oudre, Y. Grenier, and C. Févotte. Chord recognition by fitting rescaled chroma vectors to chord templates. Technical report, Telecom Paritech, 2009.

[9] J. Serrà, X. Serra, and R.G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11:093017, 2009.