

# TO CATCH A CHORUS: USING CHROMA-BASED REPRESENTATIONS FOR AUDIO THUMBNAILING

*Mark A. Bartsch*

University of Michigan  
EECS Department  
1101 Beal Avenue, 143 ATL  
Ann Arbor, MI 48109-2110  
mbartsch@eecs.umich.edu

*Gregory H. Wakefield*

University of Michigan  
EECS Department  
1101 Beal Avenue, 142 ATL  
Ann Arbor, MI 48109-2110  
ghw@eecs.umich.edu

## ABSTRACT

An important application for use with multimedia databases is a browsing aid, which allows a user to quickly and efficiently preview selections from either a database or from the results of a database query. Methods for facilitating browsing, though, are necessarily media dependent. We present one such method that produces short, representative samples (or “audio thumbnails”) of selections of popular music. This method attempts to identify the chorus or refrain of a song by identifying repeated sections of the audio waveform. A reduced spectral representation of the selection based on a chroma transformation of the spectrum is used to find repeating patterns. This representation encodes harmonic relationships in a signal and thus is ideal for popular music, which is often characterized by prominent harmonic progressions. The method is evaluated over a sizable database of popular music and found to perform well, with most of the errors resulting from songs that do not meet our structural assumptions.

## 1. INTRODUCTION

With the growing prevalence of large databases of multimedia content, the ability to quickly and efficiently browse selections from such databases is extremely important. This is especially true with advanced multimedia search and retrieval systems, where the user must be able to preview hits rapidly to determine their relevance to the original search. In order to improve the efficiency of browsing, one must consider not only the cost of delivery (in bandwidth, for instance) but also the time required to audition selections. Because of the wide variety of media that one may wish to browse, methods that facilitate such browsing must be media-dependent. The browsing of images, for instance, can be facilitated using smaller, downsampled versions of the original images. Similarly, a database that is predominantly comprised of audio recordings of speech can be well represented with text transcripts or similar summarizations. For music, we propose that a useful reduction arises from the identification of a short, representative portion of the original selection, an “audio thumbnail.”

The identification of such a representative sampling is not trivial in the general case. In classical music, a good “thumbnail” might capture the introduction of a prominent theme or motif. Identifying such an introduction is complicated by the musical context surrounding it. Much popular music, though, possesses a simpler musical form that involves the repetition of a chorus or refrain. These repeated sections of the song are typically prominent and

are generally sections that are readily recognized or remembered by the listener. Thus, if we can identify the repeated sections in a piece of popular music, we are likely to have also identified a good thumbnail. This is a somewhat simpler problem than the general case of audio thumbnailing; for this reason, in the present work we restrict our attention to popular music.

The problem of audio thumbnailing has been addressed previously by Logan and Chu [1], who developed algorithms for finding key phrases in selections of popular music. Their work focused on the use of Hidden Markov Models and clustering techniques on mel-frequency cepstral coefficients (MFCCs), a set of spectral features that have been used with great success for applications in speech processing [2]. Their system was subjectively evaluated on a relatively small selection of Beatles songs. In another work, Foote [3] identifies this problem, which he calls audio “gisting,” as an application of his measure of audio novelty. This audio novelty score is based on the similarity matrix, which compares frames of audio based on features extracted from the audio. Foote leaves details such as the similarity metric and feature class as design decisions; however, Foote does recommend the use of MFCCs as a feature class for computing audio novelty [4].

Here we present a new system that automatically generates audio thumbnails for selections of popular music. Our system employs a feature-classification framework for audio analysis. The key aspects of this system are its use of feature similarity for detecting musical recurrence and its novel feature class for representing musical structure. This feature class represents the spectrum in terms of pitch-class, and is derived from the *chromagram* [5]. In the following, we present the chromagram and our chroma-based feature class, outline the operation of our system, and present results to evaluate the performance of our system on a sizable database of diverse material.

## 2. THE CHROMAGRAM AND A CHROMA-BASED FEATURE CLASS

In the 1960’s, Shepard reported two distinct attributes of pitch perception, *tone height* and *chroma* [6]. Tone height describes the general increase in the pitch of a sound as its frequency increases. Chroma, on the other hand, is cyclic in nature with octave periodicity. Under this formulation two tones separated by an integral number of octaves share the same value of chroma. This is an intuitive concept for musicians, since chroma is closely related to the musical-theoretic concept of pitch class. Later work by Patter-

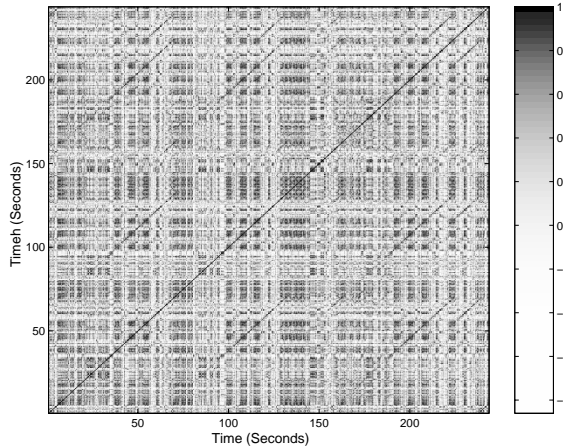


Figure 1: The similarity matrix,  $C$ , for Jimmy Buffet’s *Margaritaville*, showing the similarity between individual frames of the song.

son [7] suggested that one could decompose frequency into similar attributes.

We have found it useful to employ the musical relevance of chroma in the development of features for our structural pattern recognition. Suppose that we restructure the frequency spectrum into a *chroma spectrum* [5]. This forms the basis for the chromagram. Under such a restructuring, a harmonic series is mapped, in a many-to-one fashion, onto a relatively small number of chroma values. The first twenty harmonics of a harmonic series fall on only ten different chroma values, while thirteen of those twenty occupy only four distinct chroma values. This energy compaction is an important property of the chromagram. Furthermore, all but four of these first twenty harmonics fall within 15 cents of the closest “ideal” chroma value of the equal tempered scale. This suggests that we might discretize the chromagram into twelve distinct chroma “bins” corresponding to the twelve pitch classes without a significant loss of fidelity in the representation.

Performing this mapping procedure on the spectrum of a frame of audio data provides us with a highly reduced representation of the frame, consisting of a single twelve-element feature vector. One of the most useful properties of this feature vector is its ability to encode the harmony within a given song. Thus, two audio frames with similar harmonic content will have similar feature vectors. Of course, other aspects of the audio signal are also encoded (instrumentation, for instance, affects the chroma vector as well as can changes in the “timbre” of an instrument’s sound). With this feature vector, we can measure the similarity between two audio frames simply by measuring the correlation between their feature vectors. Then, we can further measure the correlation between extended regions by summing the correlation between their individual frames. This procedure forms the basis for our thumbnailing algorithm, which is presented in the next section.

### 3. ALGORITHM DESCRIPTION

#### 3.1. Frame Segmentation

Before the algorithm begins, we must first define a frame segmentation for the song. We have found that using a dynamic,

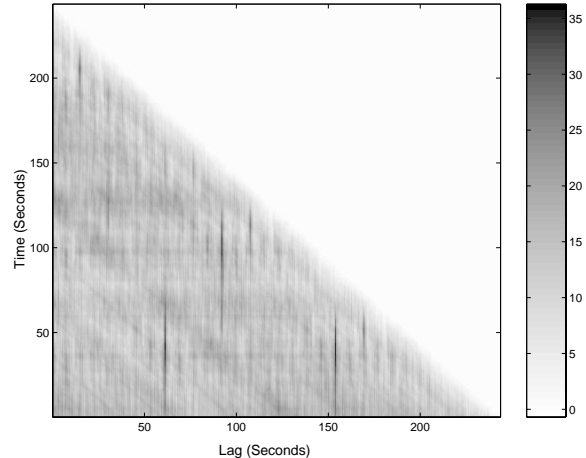


Figure 2: The time-lag surface,  $T$ , for Jimmy Buffet’s *Margaritaville*, showing the similarity between one segment of the song and a segment  $lag$  seconds ahead of it.

beat-synchronous frame segmentation improves the system’s performance. Thus, as a preprocessing step we apply a beat-tracking algorithm developed by Simon Dixon [8] to the selection under consideration. While not perfect, the beat-tracking algorithm performs well for a wide variety of popular music. Typically, the resulting frames are on the order of one-quarter to one-half of a second in length.

#### 3.2. Feature Calculation

For each frame of audio data, we compute a feature vector by calculating the logarithmic magnitude of a zero-padded DFT, labeling each DFT bin with the appropriate pitch class, and then taking the arithmetic mean of all similarly-classified bins. In the labeling process, we restrict the frequency range to 20 Hz to 2000 Hz. We have found that this frequency range provides a sufficiently rich description of the musical sounds that we are considering. We also subtract the mean of the resulting vector, which normalizes the feature vector with respect to the original signal’s amplitude.

#### 3.3. Correlation Calculation

After feature calculation, we compute the correlation between each pair of vectors calculated in step one. We place the results in a similarity matrix that has lines of constant lag oriented along the diagonals of the matrix. Thus, an extended region of similarity between two portions of a song will show up as an extended area of high correlation along one of the diagonals. One example of such a similarity matrix is shown in Figure 1.

#### 3.4. Correlation Filtering

In the next step, we filter along the diagonals of the similarity matrix to compute similarity between extended regions of the song. The size of these regions is dependent upon the length of the filter’s impulse response. We use a uniform moving average filter where the length of the impulse response is left as a design parameter of the system. The filtering results are placed in a restructured time-lag matrix, in which the lines of constant lag are oriented along the

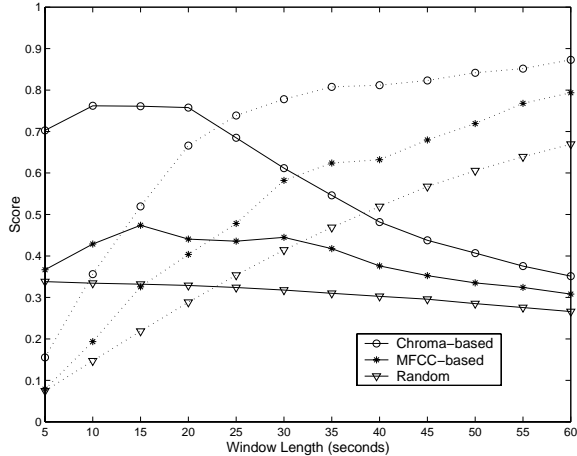


Figure 3:  $P_p$  (solid) and  $P_r$  (dotted) scores for chroma-based algorithm, MFCC-based algorithm, and random thumbnail selection.

columns of the matrix. An example of such a time-lag matrix is shown in Figure 2.

### 3.5. Thumbnail Selection

The final step of the algorithm is the selection of the thumbnail itself. This selection occurs by locating the maximum element of the time-lag matrix subject to two constraints. To prevent the selection of quick repetitions and fading repeats, we require that this location have a lag greater than one-tenth the length of the song and occur less than three-fourths of the way into the song. The thumbnail is then defined by the time-position of this maximum, which corresponds to the time that the first of the pair of sections begins, and the length of the window used for filtering.

## 4. ALGORITHM EVALUATION

A database of popular music was used to evaluate the performance of the proposed system. The database is comprised of ninety-three selections of popular music, with styles including rock, folk, dance, country-western, and numerous others. To offset the structural ambiguity of some popular music, we have also included in the database a number of contemporary Christian hymns with a very clear chorus-verse structure.

To evaluate the output of the system, it is necessary to know what portions of a song would make good thumbnails. This is accomplished by hand-selecting portions of each song as “truth” intervals. For the majority of songs, these truth intervals delimit repetitions of the chorus or refrain of the song. Not all of the songs, however, possess a single, clearly defined chorus or refrain. In such cases, we select intervals that seem to be representative of the song. In a few cases, for instance, two equally reasonable candidates for a refrain are both selected throughout the song. In others, the individual verses of the song are identified. Often these choices are somewhat arbitrary; however, we have attempted to maintain consistency as much as possible.

Two scoring methods are used to evaluate the algorithm’s output. The first score,  $P_r$ , is defined to be the length of the longest overlap between the output interval and a truth interval in the song

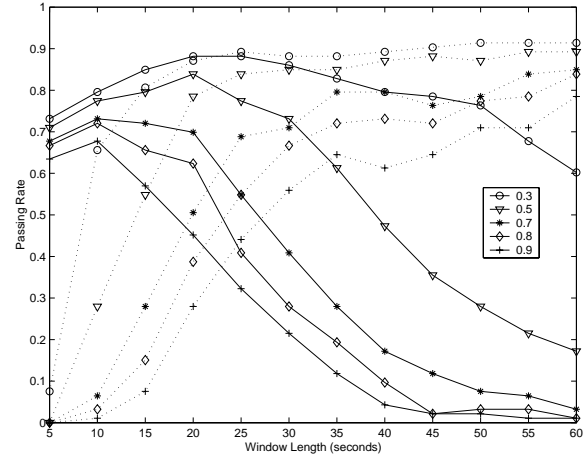


Figure 4: The chroma-based algorithm’s passing rate for  $P_p$  (solid lines) and  $P_r$  (dotted lines) under various thresholds.

divided by the total length of that truth interval. This score is effectively a frame-level recall rate. An interval receives a sub-optimal  $P_r$  score when the output does not contain all of the relevant chorus. The second score,  $P_p$ , is defined as the same overlap length divided by the length of the output interval. This score is effectively a measure of frame-level precision.  $P_p$  errors occur when the system includes selects portions of a song that are not contained in the chorus. Clearly, there is an inherent tradeoff between these scores. An output interval that is too long will most likely have a high  $P_r$  but a low  $P_p$ ; if the thumbnail is too short, the situation is reversed.

Figure 3 shows a comparison of the performance of our chroma-based algorithm versus the same algorithm using mel-frequency cepstral coefficients and also a random selection method. For each method, the mean of each score over the entire database is plotted as a function of window size. The MFCCs are calculated at the same rate as the chroma-based features; aside from the use of a different set of features, the algorithm is the same. The random selection method calculates the mean score of 1000 uniformly distributed thumbnails for each song. From this figure, we can see that our chroma-based implementation of the algorithm performs substantially better than the random selection method and the algorithm employing MFCCs for both scores and over all of the window lengths plotted here. Also, the expected tradeoff between  $P_r$  and  $P_p$  with window size is clearly shown. It is interesting to note that the intersection of all three sets of curves occurs around 23 seconds, which is the mean length of truth intervals in our database. This should not be surprising; if the output selection has the same length as the overlapping truth interval,  $P_r$  and  $P_p$  will be identical.

Another useful measure of performance is the fraction of the songs with a score exceeding some threshold, or the “passing rate.” Figure 4 displays the chroma-based algorithm’s passing rates for both scores versus window length under various thresholds. Once again, the tradeoff between the two scores is evident, and the intersection of the two score curves for each threshold lies in the vicinity of 23 seconds. This figure shows that we can obtain good passing rates for one score without too great an effect on the other if we choose a window size around 20 to 25 seconds.

It is instructive to see when and why the system fails. The

most common failure occurs when the chorus or refrain is repeated, but there is some change, either in instrumentation or in the musical structure of the repeat. Failure in these cases occurs when the verses (or some other sections) of the song are more similar to one another than the modified repetitions of the chorus or refrain. These cases violate our initial assumption of high correlation between instances of the chorus and indicate that this assumption should be relaxed. A less common error occurs when the repetition of some “uninteresting” portion of the song has a high enough correlation enough to overshadow the repetitions of the chorus. One example of this is the repetition of “Hello” in Nirvana’s *Smells Like Teen Spirit*, which is selected by our algorithm as a good thumbnail. In other cases this may be an instrumental section, such as the introduction of the song.

## 5. DISCUSSION

We have shown that our algorithm for selecting audio thumbnails operates quite well on one database of popular music. Generally, the system fails when a song does not meet our initial assumption that strongly repeated portions of a song correspond to the chorus, refrain, or otherwise important part of a song. We can generalize this cause of error and predict that the system will most likely perform poorly on types of music that do not have the simple “verse-refrain” form often found in popular music. We have already argued that the structure of classical music is too complicated to yield readily to this simple approach. Similarly, the improvisational nature of jazz and blues, for instance, violates our original assumption as well. Thus we would not expect the system to perform well in those cases.

The most important conclusion that can be drawn from these results relates to the potential of chroma-based representations for encoding musical structure. While the algorithm does perform better than chance when using MFCCs as features, the algorithm performs significantly better when we employ our chroma-based features. This indicates that these chroma features are a much better choice for this application. MFCCs are a general purpose measure of the smoothed spectrum of an audio signal which primarily represent the timbral aspects of the sound. Since we are seeking structure in music, we would prefer to represent the elements of the music that provide its structure: notes, harmonic progressions, and so on. These elements are encoded in the fine spectral structure of the audio signal and are not well represented by MFCCs. Our chroma-based representation retains and exploits much of this fine structure.

We have hypothesized that a chroma-based feature class can be used to capture harmonic relationships within a song. An examination of the chroma features themselves indicates that they do encode harmonic relationships to some extent. The features exhibit long-term patterns that are indicative the dominant harmonies in a song and may be useful for determining musical key. Locally, one can identify changes that map to harmonic progressions in the song. This suggests that a chroma-based feature class could be useful for any application that requires structural analysis of music based on harmonic relationships.

There are a number of interesting extensions that could be made to this simple method. First, the ability to optimize over window size would be useful to mitigate the tradeoff between the two sources of error. Further, performance would likely be improved if the system could take advantage of more structural information than just the highest pairwise correlation. This extension could

also form the basis of an extension to a full music-segmentation system based on structural content, which could be valuable for musical form analysis. It is also possible that this method would be a useful technique for multimedia search and retrieval systems.

## 6. CONCLUSION

We have presented a system which uses chroma-based representations of sound to isolate repeated sections within popular music for the purpose of producing short, representative samples of entire songs. Such a system has numerous applications, including the browsing of musical databases and multimedia search results. Perhaps more importantly, the success of this system serves to illustrate the potential of chroma-based representations for the structural analysis of musical content as an alternative to mel-frequency cepstral coefficients. This system provides a first step towards using chroma-based representations as an important element of more sophisticated analysis systems, including segmentation and search-and-retrieval.

## 7. ACKNOWLEDGEMENTS

This material is based upon work supported, in part, by a National Science Foundation Graduate Research Fellowship and by grants from the National Science Foundation (IIS-0085945) and the MusEn Project at the University of Michigan through the Office of the Vice President for Research. The authors would like to thank members of the MusEn Project, William Birmingham, Bryan Pardo, Colin Meek, and Maureen Melody, for their comments and contributions to this work. The authors would also like to thank the anonymous reviewers of this paper for their helpful suggestions for improvements.

## 8. REFERENCES

- [1] B. Logan and S. Chu, “Music summarization using key phrases,” in *International Conference on Acoustics, Speech and Signal Processing*, 2000.
- [2] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.
- [3] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” in *Proceedings of IEEE International Conference on Multimedia and Expo*, 1999, vol. I, pp. 452–455.
- [4] J. Foote, “Visualizing music and audio using self-similarity,” in *Proceedings of ACM Multimedia '99, Orlando, Florida*, November 1999, pp. 77–80.
- [5] G. H. Wakefield, “Mathematical representation of joint time-chroma distributions,” in *SPIE, Denver, Colorado*, 1999.
- [6] R. Shepard, “Circularity in judgements of relative pitch,” *Journal of the Acoustical Society of America*, vol. 36, pp. 2346–2353, 1964.
- [7] R. D. Patterson, “Spiral detection of periodicity and the spiral form of musical scales,” in *Psychology of Music*, chapter 14, pp. 44–61. 1986.
- [8] S. Dixon, “A lightweight multi-agent musical beat tracking system,” in *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*. 2000, AAAI Press.